

---

**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: Mechatronika

**Digitální syntetická impedance pro výzkum  
v oblasti tlumení vibrací**

**Digital synthetic impedance for research of  
vibration damping**

**Diplomová práce**

Autor: **Bc. Jiří Budasz**

Vedoucí práce: Ing. Jan Václavík

Konzultant: Ing. Martin Černík, Ph.D., Ing. Pavel Márton, Ph.D.

V Liberci 2014



Místo této strany se vloží podepsané zadání.



## **Prohlášení**

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 - školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiju-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis



## Abstrakt

Práce se zabývá návrhem obvodu pro implementaci digitální syntetické impedance s absolutní hodnotou od  $10\ \Omega$  do  $0,1\ \text{M}\Omega$  ve frekvenčním rozsahu 50 Hz až 5 kHz. Jsou uvedeny a diskutovány možnosti realizace takového obvodu. Zvolená topologie se skládá z analogové části s operačními zesilovači a z digitální, ve které je použit mikrokontrolér ARM. Vybrané řešení obvodu je analyzováno z teoretického hlediska. Analýza zahrnuje především návrh digitálních filtrů pro implementaci. Práce popisuje praktickou realizaci obvodu, firmware mikrokontroléru a skripty MATLABu použité pro teoretickou rozvahu, návrh filtrů a komunikaci s obvodem. Správná funkce zařízení je ověřena měřením syntetizovaných impedancí zadaných jak náhradním obvodem, tak i tabulkou průběhů absolutní hodnoty a fáze v závislosti na frekvenci. Změřené charakteristiky impedancí se s dostatečnou přesností shodují s teoretickými.

Klíčová slova: Aktivní tlumení vibrací, bočníkové tlumení, syntetická impedance, negativní kapacita.



## Abstract

This thesis deals with design of a circuit, which implements digital synthetic impedance between  $10\ \Omega$  and  $0.1\ \text{M}\Omega$  in a frequency range from 50Hz to 5kHz. Several possibilities of realization of such circuit are presented and discussed. Chosen solution consists of an analogue and digital part, where the impedance synthesis is realized by an ARM microcontroller. The circuit topology is analyzed from the theoretical perspective and included in design of digital filters. The thesis further describes practical realization of the circuit, the firmware for the microcontroller and the set of MATLAB scripts for complete development of digital filters and communication with the circuit. Correct function of the realized device is verified by measurement of the synthesized impedances defined by i) several equivalent circuits or by ii) tables of absolute value and phase of impedance in a given frequency range. Measured characteristics are found to be in a very good agreement with theoretical predictions.

Keywords: Active vibration damping, shunt damping, synthetic impedance, negative capacity.



## Obsah

|   |    |
|---|----|
| Prohlášení.....   | 3  |
| Abstrakt.....   | 4  |
| Abstract .....  | 5  |
| Obsah .....   | 6  |
| Seznam obrázků .....  | 8  |
| Seznam grafů .....  | 9  |
| Seznam tabulek .....  | 9  |
| 1 Úvod .....  | 10 |
| 2 Požadavky a možnosti realizace digitální syntetické impedance ..... | 12 |
| 2.1 Požadavky na syntetizované impedance .....                        | 12 |
| 2.2 Obvodový simulátor s proudovými konvejory .....                   | 14 |
| 2.3 Obvod syntetické admitance .....                                  | 18 |
| 2.4 Výběr koncepce pro syntetizaci impedance .....                    | 20 |
| 3 Teoretická analýza a volba algoritmů .....                          | 22 |
| 3.1 Elektrická analýza .....  | 22 |
| 3.2 Analýza části zpracování signálu .....                            | 25 |
| 3.3 Diskretizační metody .....  | 29 |
| 3.4 Diskretizace základních impedancí .....                           | 32 |
| 3.5 Návrh filtru z tabulky hodnot .....                               | 35 |
| 4 Hardwarové řešení .....   | 37 |
| 4.1 Digitální část .....  | 37 |
| 4.2 Rozhraní analogové a digitální části .....                        | 38 |
| 4.3 Analogová část .....  | 40 |



|     |  |    |
|-----|--|----|
| 5   | Programové vybavení .....                                  | 42 |
| 5.1 | Firmware mikropočítače .....                               | 42 |
| 5.2 | Program v počítači .....                                   | 46 |
| 6   | Experimentální ověření syntézy .....                       | 50 |
| 6.1 | Ověření základní funkce obvodu .....                       | 50 |
| 6.2 | Měření základních impedancí .....                          | 53 |
| 6.3 | Ověření návrhu filtru z tabulky hodnot.....                | 55 |
| 7   | Dosažené výsledky a možnosti pokračování práce .....       | 62 |
| 8   | Závěr.....   | 63 |
|     | Použitá literatura .....                                   | 64 |
|     | Příloha A Obsah přiloženého CD.....                        | 66 |
|     | Příloha B Odvození diskretizační metody mapování pólů..... | 67 |
|     | Příloha C Diskretizace základních impedancí.....           | 71 |
|     | Příloha D Schéma digitálního modulu .....                  | 74 |
|     | Příloha E Schéma modulu převodníků.....                    | 75 |
|     | Příloha F Schéma analogového výstupního modulu .....       | 77 |
|     | Příloha G Výpisy skriptů v MATLABu .....                   | 78 |
|     | Příloha H Měření základních impedancí.....                 | 98 |



## Seznam obrázků

|   |    |
|---|----|
| Obr. 1.1 Blokové schéma nové koncepce bočníkového tlumení vibrací.....  | 10 |
| Obr. 2.1 Základní vybrané typy impedancí pro Laboratoř tlumení vibrací.....   | 12 |
| Obr. 2.2 Blokové schéma <i>univerzálního obvodového simulátoru</i> .....  | 14 |
| Obr. 2.3 Náhradní elektrické schéma ideálního proudového konveje druhé generace .   | 15 |
| Obr. 2.4 Dvě verze vstupně-výstupního uzlu .....  | 16 |
| Obr. 2.5 Náhradní elektrické schéma simulátoru a detailní struktura obvodového simulátoru použitého Giannim aj. (2007)..... | 17 |
| Obr. 2.6 Náhradní elektrické schéma <i>obvodu syntetické admitance</i> a jeho obvodová implementace .....                   | 18 |
| Obr. 2.7 Schéma syntetizované impedance .....   | 19 |
| Obr. 3.1 Elektrické schéma pro analýzu .....  | 22 |
| Obr. 3.2 Blokové schéma části pro zpracování signálu.....   | 25 |
| Obr. 4.1 Blokové schéma hlavních funkcí digitální části .....   | 37 |
| Obr. 4.2 Zjednodušené schéma analogového výstupního modulu .....  | 40 |
| Obr. 4.3 Foto použitého hardware .....  | 41 |
| Obr. 5.1 Vývojový diagram firmware mikropočítače .....  | 44 |
| Obr. 6.1 Průběhy napětí a proudu syntetizované impedance.....   | 50 |





## Seznam grafů

|  |    |
|--|----|
| Graf. 2.1 Průběhy absolutní hodnoty a fáze impedance v závislosti na frekvenci ..... | 13 |
| Graf. 3.1 Vliv vzorkovací frekvence na průběh fáze syntetizované impedance .....     | 28 |
| Graf. 3.2 Vliv diskretizační metody na průběh amplitudy a fáze .....                 | 33 |
| Graf. 6.1 Průběhy absolutních hodnot syntetizovaných odporů .....                    | 51 |
| Graf. 6.2 Měření průběhu fáze činného odporu .....                                   | 52 |
| Graf. 6.3 Měření impedančních průběhů sériového RLC obvodu .....                     | 53 |
| Graf. 6.4 Charakteristiky navrženého filtru .....                                    | 56 |
| Graf. 6.5 Srovnání průběhu absolutních hodnot různých přenosů $F(z)$ .....           | 57 |
| Graf. 6.6 Porovnání aproximace pomocí přenosu FIR a IIR dle optimalizace .....       | 59 |
| Graf. 6.7 Měření aproximace dat IIR filtrem, po úpravě tvaru zadání .....            | 61 |

## Seznam tabulek

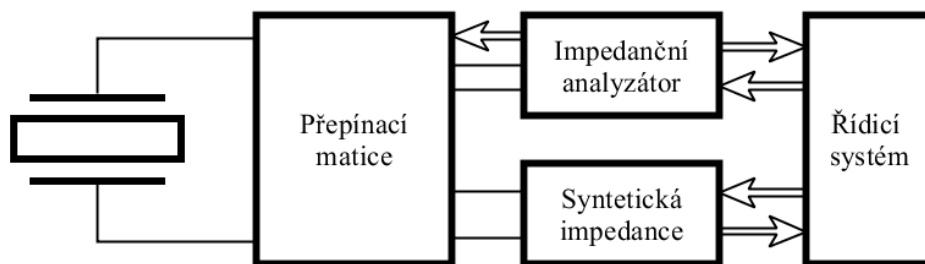
|   |    |
|---|----|
| Tab. 4.1 Technické parametry použitého D/A převodníku .....               | 39 |
| Tab. 4.2 Technické údaje použitého A/D převodníku .....                   | 39 |
| Tab. 5.1 Orientační časy zpracování dat .....                             | 43 |
| Tab. 5.2 Přehled naprogramovaných ovládacích a signalizačních prvků ..... | 45 |



## 1 Úvod

Vibrace jsou u většiny zařízení v technické praxi vnímány jako parazitní jev, zároveň jsou ale jevem velmi častým. Negativně ovlivňují funkci i stav zařízení. Proto jsou rozvíjeny mnohé metody pro tlumení vibrací. Jedním z přístupů využívajících piezoaktuátory je tzv. bočníkové tlumení, anglicky označováno zkratkou PSD (Piezoelectric Shunt Damping). Tlumící piezočlen je připojen mezi vibrující těleso a hmotu, kterou je nutné izolovat od vibrací. Piezočlen slouží zároveň jako snímač i aktuátor. K tlumicímu piezočlenu je připojen elektronický obvod o určité impedanci. Tlumících účinků lze dosáhnout připojením impedance vytvořené z pasivních součástek. Lepších výsledků je možné dosáhnout s využitím aktivního obvodu. Pak je impedance syntetizována aktivním elektronickým obvodem. Díky tomu lze například efektivněji potlačovat rezonance dané struktury. V závislosti na parametrech impedance lze řídit tuhost tlumícího piezočlenu. V extrémním případě teoreticky nulové tuhosti jsou vibrace úplně potlačeny.

Jak ukazuje Kodejška (2013) ve svém experimentu, maximálního útlumu vibrací je dosaženo v okamžiku, kdy absolutní hodnoty elektrických impedancí piezočlenu a syntetizované impedance jsou ve správném poměru daném nastavením obvodu. Hlavní vliv na útlum má přesnost hodnot fázové charakteristiky. Jak uvádí Kodejška (2013), maximální útlum je při nejlepší shodě fázových charakteristik a naopak i malý rozdíl fází útlumu brání. Úskalím této metody je skutečnost, že piezočlen v průběhu času mění své vlastnosti v závislosti na teplotě, a tak je nutno měnit i připojenou impedanci. Je rovněž nutné dosáhnout relativně vysoké přesnosti syntetizované impedance až pod 1 % ve vybraném frekvenčním rozmezí.



Obr. 1.1 Blokové schéma nové koncepce bočníkového tlumení vibrací



V současné době jsou taková aktivní zařízení studována na Fakultě mechatroniky, informatiky a mezioborových studií v čistě analogové podobě. Případně jsou doplněna o digitální část, která pouze doladňuje analogový obvod tak, aby se přizpůsobil změnám vlastností piezočlenu. Nevýhodou tohoto přístupu je, že jedno zařízení může syntetizovat pouze jednu jednoduchou strukturu impedance. Vznikla tak potřeba zcela nové koncepce, jejíž základní princip zachycuje blokově obr. 1.1. Tlumící piezočlen je připojen k přepínací matici, která umožňuje propojení jednak s obvodem syntetické impedance a jednak s impedančním analyzátelem. Ten měří parametry elektrické impedance jak piezočlenu tak i syntetizačního obvodu. Všechna data pak předává nadřazenému řídicímu systému, který podle vhodného algoritmu navrhne korekci, případně úplně nový průběh syntetizované impedance. Tuto informaci řídicí systém předává obvodu syntetické impedance. Popsaný přístup by měl vyřešit problém teplotní a časové nestálosti parametrů piezočlenu a také dosáhnout požadované přesnosti shody syntetizované impedance s impedancí piezočlenu. Z výše popsané koncepce vychází vize týmu Laboratoře tlumení vibrací na vytvoření zařízení schopného praktického nasazení.

Práce řeší návržení, teoretickou analýzu a vytvoření jedné části výše popsané struktury. Řeší zařízení, které vzhledem ke dvěma svorkám ( $\hat{U}$  představuje napětí na svorkách a  $\hat{I}$  proud tekoucí do zařízení) syntetizuje požadovanou impedanci  $\bar{Z}$  dle vztahu:

$$\bar{Z} = \frac{\hat{U}}{\hat{I}} \quad (1.1)$$

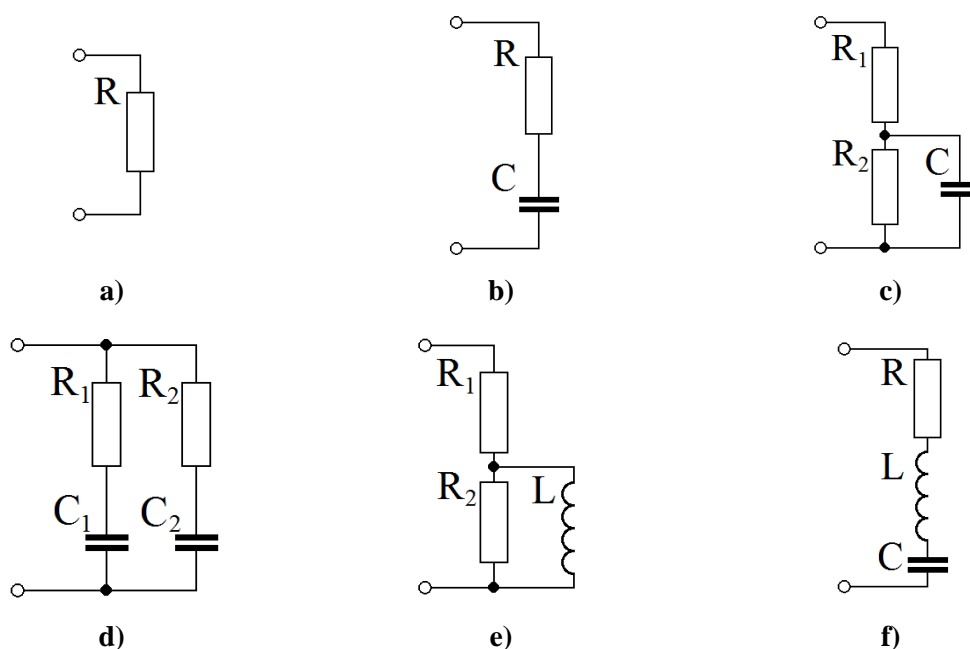
To vše ve frekvenčním rozsahu od 50 Hz do 5 kHz a v rozsahu absolutní hodnoty impedance od 10  $\Omega$  do 0,1 M $\Omega$  za předpokladu harmonického buzení a linearitu obvodu. Zařízení by mělo být schopno syntetizovat jak impedance zadané pomocí vzorce s náhradním schématem a hodnotami základních pasivních prvků, tak i impedance, které nelze takto popsat a jejichž parametry jsou určeny pouze tabulkou hodnot absolutní hodnoty a fáze. Toto je zároveň jeden z hlavních důvodů, proč musí být jádro zařízení digitální. Zadané hodnoty impedance jsou předávány z nadřazeného řídicího systému standardním rozhraním. Vzhledem k tomu, že výsledné zařízení i celá sestava jsou určeny pro použití v laboratoři, bude řídicím systémem osobní počítač.



## 2 Požadavky a možnosti realizace digitální syntetické impedance

### 2.1 Požadavky na syntetizované impedance

Základní požadavky na obvod digitální syntetické impedance jsou v zadání práce, tedy frekvenční rozsah od 50 Hz do 5 kHz a absolutní hodnota impedance v rozsahu od  $10\ \Omega$  do  $0,1\ \text{M}\Omega$ . V řešení bylo nutné nejprve zjistit, jaké typy impedancí jsou potřeba syntetizovat pro danou úlohu v Laboratoři tlumení vibrací. Tedy pro systém, ve kterém je kmitající hmota upevněna přes piezoelektrický aktuátor ke zdroji vibrací. Ten je spojen s rámem, resp. mechanickou zemí. Cílem je dosáhnout nejmenšího, ideálně nulového, přenosu vibrací ze zdroje na těleso. Jak ukazuje odvození provedené Slukou (2007), nejdůležitějším prvkem k dosažení tohoto cíle je syntetizace obvodů s kapacitorem majícím zápornou hodnotou. Sluka (2007) provedl experiment, ve kterém tuto schopnost prokázal s použitím analogového obvodu syntetizujícího chování záporného kapacitoru.



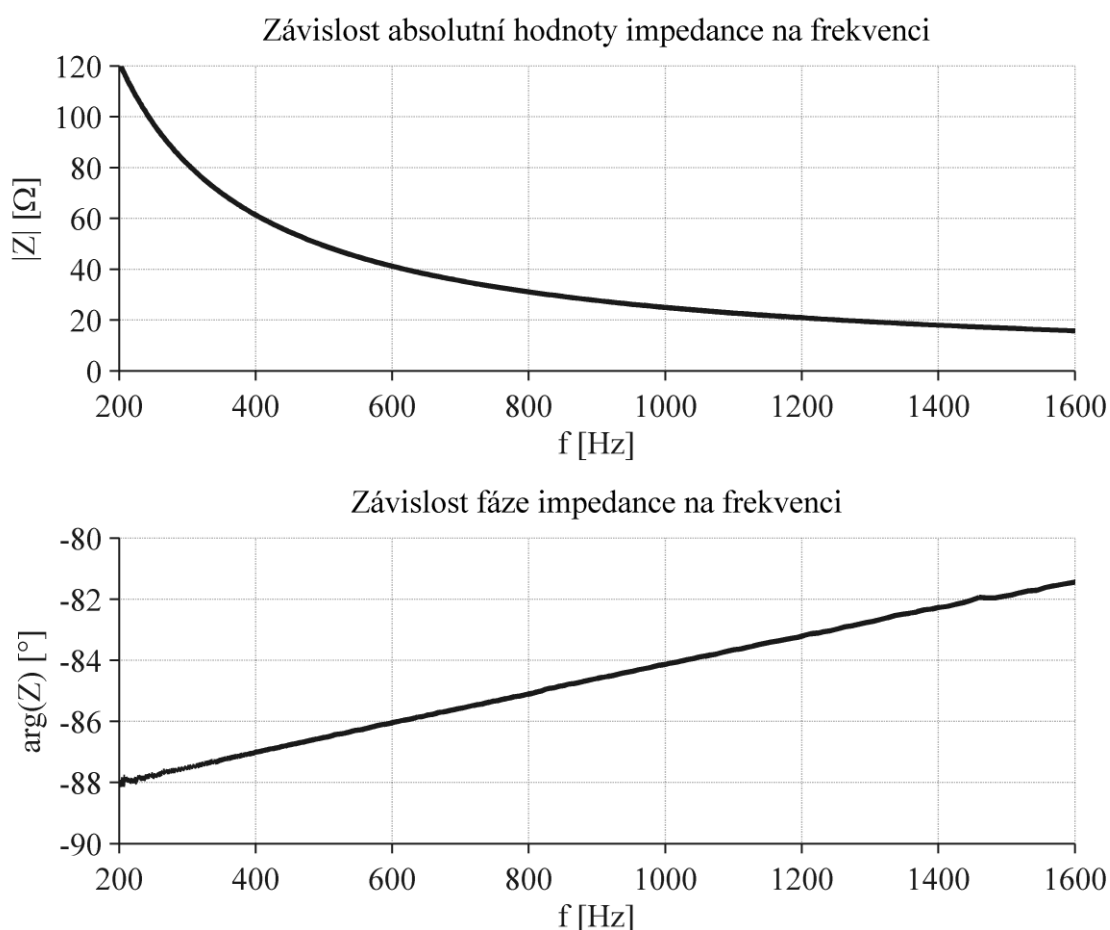
**Obr. 2.1** Základní vybrané typy impedancí pro Laboratoř tlumení vibrací

Schopnost navrhovaného systému syntetizovat impedance bude ověřena na několika základních typech zachycených na obr. 2.1. Základem je nejjednodušší impedance, tedy činný odpor, obr. 2.1 a). Z výše uvedeného důvodu následuje několik obvodů kombinujících činný odpor s kapacitorem, obr. 2.1 b) až d), jak s hodnotou kapacity kladnou tak především zápornou. Právě tyto kombinace budou pravděpodobně



nejčastěji používány s tím, že na požadované frekvenci bude reaktance kapacitoru převažovat. Jak uvádí Gianni aj. (2007), jsou v této oblasti tlumení vibrací používána i zapojení s induktory. Proto je zařazeno i kombinované zapojení odporů s induktorem dle obr. 2.1 e). Jak uvádí Fleming aj. (2002), součástí impedancí v tlumení vibrací jsou v určitých případech i rezonanční obvody. Z tohoto důvodu je zařazen sériový rezonanční RLC obvod na obr. 2.1 f).

Všechny výše popisované typy impedancí byly zadány náhradním el. schématem a hodnotami základních prvků nebo vzorcem impedance. Dalším typem, který by měl být navrhovaným systémem syntetizován, je průběh zadáný pouze tabulkou hodnot průběhu absolutní hodnoty a fáze v závislosti na frekvenci, ať již lze průběhy jednoduše popsat analyticky nebo nikoliv. Příkladem jsou data pocházející z experimentu Kodejšky (2013), provedeného v Laboratoři tlumení vibrací. Průběhy zachycuje graf. 2.1.

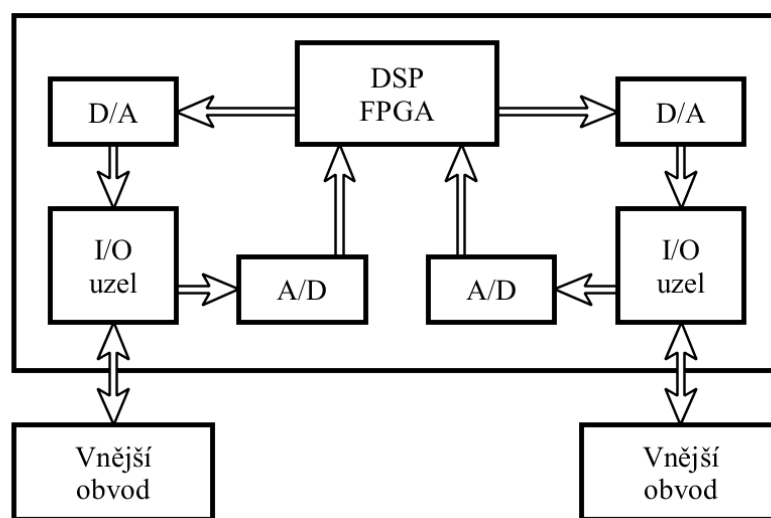


**Graf. 2.1** Průběhy absolutní hodnoty a fáze impedance v závislosti na frekvenci, data od Kodejšky (2013)



## 2.2 Obvodový simulátor s proudovými konvejory

Pomocí řešerše bylo také nutné zjistit možnosti digitální syntetizace impedance. Jednou variantou je použití hybridního obvodu s proudovými konvejory. Realizaci i vlastnosti takového *univerzálního obvodového simulátoru* popisuje Gianni aj. (2007). Principiální blokové schéma je zachyceno na obr. 2.2. Srdcem zařízení je blok digitálního zpracování signálu (DSP). V realizaci popsané Giannim aj. (2007) je použit obvod programovatelných hradlových polí, známější pod zkratkou FPGA. Tento obvod provádí veškeré výpočty a zpracování dat, které získává od připojených A/D převodníků.



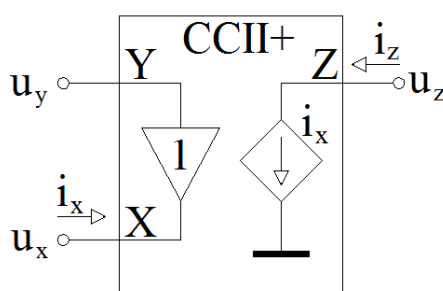
Obr. 2.2 Blokové schéma *univerzálního obvodového simulátoru*, převzato od Gianniho aj. (2007)

Výsledky výpočtu jsou předávány D/A převodníkům. FPGA obvod je připojen k osobnímu počítači pomocí USB rozhraní. Díky tomu lze konfigurovat parametry výpočtu a tím i parametry syntetizovaného obvodu. K A/D a D/A převodníkům jsou připojeny dva vstupně-výstupní uzly. Tyto uzly zajišťují propojení s vnějším analogovým obvodem, zároveň vůči těmto dvěma uzlům je syntetizována impedance daného obvodu. Hlavní součástí každého vstupně-výstupního uzlu je proudový konvektor.

Vzhledem k tomu, že proudový konvektor není často používaná součástka a pro popisovanou metodu syntetizace impedance je zásadní, věnuji část popisu uvažovanému typu proudového konvektoru. V realizaci popsané Giannim aj. (2007) je to tzv. proudový konvektor druhé generace, označovaný anglickou zkratkou CCII+ (Current Conveyor 2<sup>nd</sup> Generation). Náhradní schéma ideálního proudového konvektoru tohoto typu je na obr. 2.3. Konvektor má jeden vstup Y a dva výstupy X a Z. Tento systém značení vstupů



a výstupů není obvyklý, ale jedná se patrně o konvenci v oboru proudových konvektorů. Jak popisuje Gianni aj. (2007), mezi vstupem Y a výstupem X je napěťový sledovač tzn., že napětí  $u_x$  na výstupu X je rovno napětí  $u_y$  na vstupu Y, dle rovnice (2.1). Všechna napětí  $u_x$ ,  $u_y$  a  $u_z$  jsou vztažena k nulovému potenciálu, pro přehlednost nejsou na obr. 2.3 uvedeny šipky vyznačující tato napětí. Proud  $i_x$  tekoucí do výstupu X je zrcadlen pomocí řízeného proudového zdroje na výstup Z, což znamená, že proud  $i_z$  tekoucí do tohoto vstupu je stejný jako  $i_x$  dle rovnice (2.2). Znaménko + ve zkratce CCII+ znamená, že smysl proudu  $i_x$  je shodný se smyslem proudu  $i_z$ , přesně dle obr. 2.3.

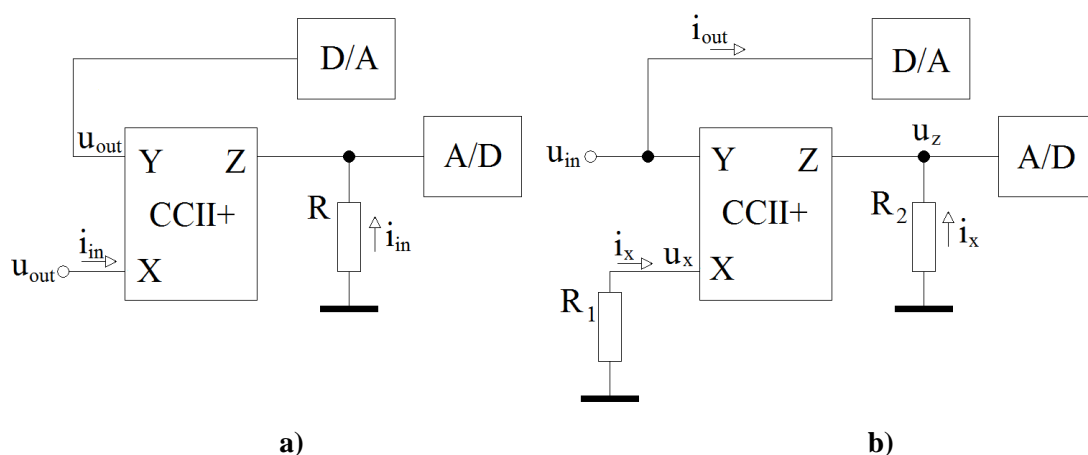


**Obr. 2.3 Náhradní elektrické schéma ideálního proudového konvektoru druhé generace, převzato od Gianniho aj. (2007)**

$$u_x = u_y \quad (2.1)$$

$$i_z = i_x \quad (2.2)$$

Díky výše popsaným vlastnostem je možné konstruovat vstupně-výstupní uzel ve dvou variantách. V první je syntetizačním zařízením měřen proud tekoucí do uzlu, napětí uzlu proti nulovému potenciálu je pak určeno D/A převodníkem na základě výpočtu. Detail tohoto provedení je na obr. 2.4 a). Vstupní proud  $i_{in}$  teče do výstupu konvektoru X. Následně je zrcadlen do výstupu Z. Vstupní odpor A/D převodníku se předpokládá nekonečný, takže stejný proud  $i_{in}$  protéká rezistorem  $R$ . Tím dochází k přeměně vstupního proudu na napětí. Je tak možné měřit A/D převodníkem s napěťovým vstupem vstupní proud  $i_{in}$ . D/A převodník po obdržení výsledku výpočtu vygeneruje výstupní napětí  $u_{out}$ , které je přivedeno na vstup konvektoru Y a napěťový sledovač mezi vstupem Y a výstupem X konvektoru zajistí, že napětí  $u_{out}$  se objeví na výstupu X a tím i na uzlu obvodu. Ve druhé verzi zachycené na obr. 2.4 b) je naopak měřeno napětí uzlu a zařízením na základě výpočtu určen proud. Vstupní napětí  $u_{in}$  je přeneseno ze vstupu konvektoru Y na výstup X. Proud  $i_x$  do výstupu X je pak určen vztahem (2.3) a zároveň je



**Obr. 2.4 Dvě verze vstupně-výstupního uzlu dle Gianniho aj. (2007), a) snímán proud, generováno napětí, b) snímáno napětí, generován proud**

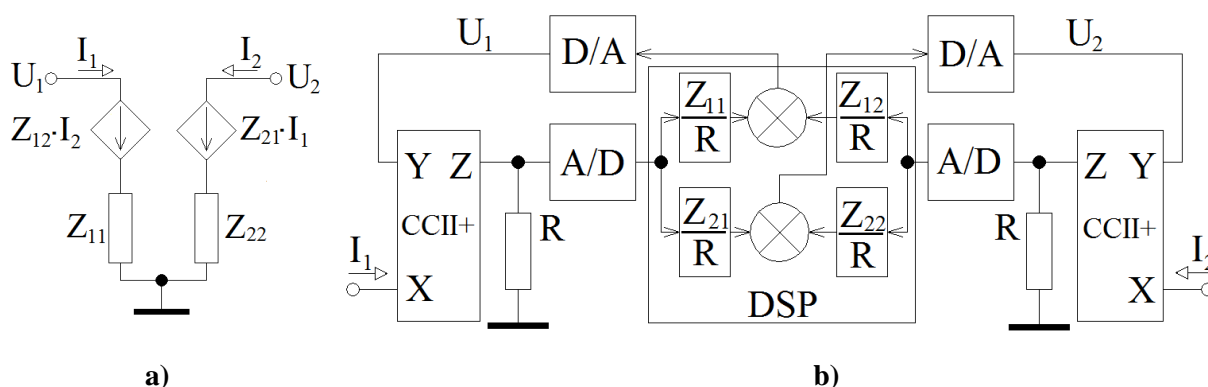
$$i_x = \frac{u_{in}}{R_1} \quad (2.3)$$

$$u_z = i_x \cdot R_2 \quad (2.4)$$

přenesen na výstup Z pomocí řízeného zdroje proudu v konveju. Díky tomu teče odporem  $R_2$  stejný proud  $i_x$ . A/D převodník poté změří napětí určené dle vztahu (2.4). Výsledek výpočtu je předán do D/A převodníku, který má v této verzi proudový výstup. Tím je určen proud  $i_{out}$  tekoucí do uzlu z okolního obvodu. Proud do vstupu konveju Y je ideálně nulový, protože Y je zároveň vstupem napěťového sledovače.

Výše popisovaný *univerzální obvodový simulátor* lze konstruovat pro různý počet vstupně-výstupních uzlů. Nicméně je třeba uvážit, že pro každý takový uzel je potřeba jeden A/D převodník, D/A převodník a proudový konveju. Lze též navrhnout uzly, které by bylo možné přepínat mezi snímáním proudu tekoucího do uzlu a snímáním napětí uzlu. Jediná komplikace takového přepínání může být v nutnosti měnit typ výstupu D/A převodníku z napěťového na proudový. Gianni aj. (2007) popisovaný simulátor použili k syntetizaci jednobranu, u kterého ani jedna ze svorek nebyla připojena k nulovému potenciálu. Detailní konfigurace simulátoru je zachycena na obr. 2.5 b). Oba vstupně-výstupní uzly jsou zapojeny tak, že snímají proud tekoucí do nich a určují napětí daných uzlů. Na základě změřených hodnot proudů jsou vypočtena výstupní napětí obou uzlů. V bloku zpracování signálu (DSP) je schematicky naznačen výpočet skládající se ze čtyř operací násobení a dvou operací sčítání. Vzhledem k tomu, že digitální zpracování signálu je prováděno pomocí obvodu FPGA, je výpočet velmi rychlý a dovoluje použití





**Obr. 2.5 a) Náhradní elektrické schéma simulátoru, b) detailní struktura obvodového simulátoru použitého Giannim aj. (2007)**

relativně vysoké vzorkovací frekvence v řádech stovek kHz. Náhradní elektrické schéma, pomocí kterého lze matematicky modelovat chování takto konfigurovaného simulátoru, je na obr. 2.5 a). Jedná se v podstatě o zapojení dvojbranu popsaného impedanční maticí s tím, že jedna svorka od každé brány je připojena k nulovému potenciálu. Zbylé dvě svorky poté představují uzly obvodu, kde je připojen simulátor. Nezávisle proměnnými veličinami jsou vstupní proudy uzlů  $I_1$  a  $I_2$ , závisle proměnnými jsou pak napětí na těchto uzlech, tj.  $U_1$  a  $U_2$ . Jejich velikost je určena těmito vztahy:

$$U_1 = Z_{11} \cdot I_1 + Z_{12} \cdot I_2 \quad (2.5)$$

$$U_2 = Z_{21} \cdot I_1 + Z_{22} \cdot I_2 \quad (2.6)$$

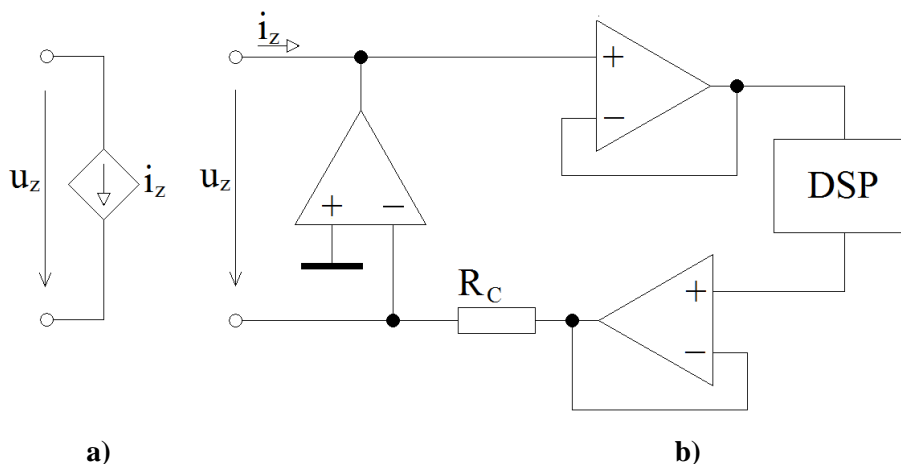
Ve vztazích (2.5) a (2.6) se již nevyskytuje hodnota odporu  $R$ , na rozdíl od schématu výpočtu na obr. 2.5 b). To je způsobeno faktem, že A/D převodníky mají napěťový vstup, a tak je potřeba převést příslušný měřený proud na napětí pomocí odporu  $R$ . Pro získání hodnoty vstupního proudu je následně nezbytné změřenou hodnotu napětí vydělit velikostí odporu  $R$ . Tudíž je první operace s hodnotami proudů doplněna dělením příslušného koeficientu odporem  $R$ .

Gianni aj. (2007) použili úspěšně výše popsaný simulátor s oběma variantami vstupně-výstupních uzlů. Syntetizovali sériový LR obvod mezi dvěma uzly, z nichž ani jeden nebyl spojen s nulovým potenciálem. Využili ho i k syntetizaci dolnoproustního filtru konečné odezvy čtvrtého řádu a pásmové propusti stejného řádu. Konfigurace simulátoru se v tomto případě ale patrně lišila. Zřejmě byl v případě obou filtrů využit jen jeden vstupně-výstupní uzel a druhým pólem impedance byl nulový potenciál obvodu. Z výsledků experimentů provedených Giannim aj. (2007) vyplývá, že výše popsaný princip je schopen úspěšně syntetizovat požadované průběhy impedance.



## 2.3 Obvod syntetické admitance

Syntetickou impedanci vzhledem ke dvěma uzlům obvodu je také možné realizovat přístupem, který popsal Fleming aj. (2000) a část popisu doplňuje Fleming (2004). Název *obvod syntetické admitance* uvádí Fleming (2004). I když je tento název možná trochu zavádějící, pochází označení z funkčního principu, který je popsán níže. Náhradní elektrické schéma je zachyceno na obr. 2.6 a). Na základě změřeného napětí mezi svorkami  $u_z(t)$  určuje proud  $i_z(t)$  mezi nimi podle vztahu (2.7). Jak uvádí Fleming aj. (2000), pokud lze považovat  $i_z(t)$  za výsledek lineární přenosové funkce  $u_z(t)$ , je možné vyjádřit vztah v Laplaceově oblasti, jak je to ve vztahu (2.8).  $U_z(s)$  a  $I_z(s)$  jsou obrazy  $u_z(t)$  respektive  $i_z(t)$ . Veličina  $Y(s)$  představuje syntetizovanou admitanci. Ta je převrácenou hodnotou syntetizované impedance  $Z(s)$  dle vztahu (2.9).



**Obr. 2.6 a) Náhradní elektrické schéma *obvodu syntetické admitance*, b) jeho obvodová implementace, převzato od Fleminga aj. (2000) a je součástí patentu od Behrense aj. (2003)**

$$i_z(t) = f(u_z(t)) \quad (2.7)$$

$$I_z(s) = Y(s) \cdot U_z(s) \quad (2.8)$$

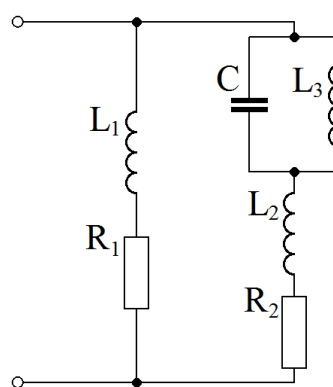
$$Y(s) \equiv \frac{1}{Z(s)} \quad (2.9)$$

Obvodová implementace řízeného zdroje je na obr. 2.6 b). Základem jsou tři operační zesilovače (OZ) a blok digitálního zpracování signálu (DSP). Podle Fleminga (2004) tvoří nejspodnější OZ, odpor  $R_C$  a OZ vlevo, bez zpětné vazby, napětím řízený zdroj proudu. Blok DSP implementuje přenosovou funkci syntetizované admitance  $Y(s)$ . To je hlavní důvod, proč Fleming (2004) nazývá tento přístup syntetickou admitancí, i když zadané průběhy jsou většinou impedanční. Vzhledem ke vztahu (2.9) mezi



impedancí  $Z(s)$  a admitancí  $Y(s)$  lze obě veličiny vzájemně jednoduše převádět. V praktické realizaci Fleming aj. (2000) uvádí, že na místě bloku DSP bylo použito hotové řešení dSPACE DSP. Nicméně jistě přichází v úvahu realizovat zapojení v podstatě s jakýmkoliv systémem, který je schopen implementovat požadovanou přenosovou funkci. Jak uvádí Fleming (2004), je možné DSP blok nahradit dokonce analogovým zapojením. Což sice není cíl, ke kterému směřuje tato práce, ale svědčí to o skutečnosti, že blok DSP lze realizovat širokým spektrem přístupů.

Z rešerše vyplynulo, že *obvod syntetické admitance* byl úspěšně použit při několika experimentech. V jednom z nich, který popisuje Fleming aj. (2002), bylo cílem tlumit vibrace soustavy, jejíž mechanická frekvenční charakteristika obsahovala dvě rezonance (dvě rezonanční maxima). Konkrétně se jednalo o tlumení kmitů kovového pásku volně podepřeného na obou koncích. K tomuto účelu bylo nutné syntetizovat impedanci, jejíž elektrické schéma je zachyceno na obr. 2.7. Vyobrazenou obvodovou strukturu sice nebude v nejbližší době nutné realizovat, protože v Laboratoři tlumení vibrací je používána jiná experimentální sestava. Důležité je ale to, že se podařilo s její syntetizací úspěšně utlumit obě rezonanční převýšení. Výsledek experimentu popsaného Flemingem aj. (2002) tak prokazuje schopnost *obvodu syntetické admitance*, který je použit jako část komplexního systému, realizovat i složitější obvodové struktury. Konkrétně struktury, které obsahují více cívek a kondenzátorů, a tudíž mají i své rezonance.



**Obr. 2.7 Schéma syntetizované impedance v experimentu popsaném Flemingem aj. (2002)**



## 2.4 Výběr koncepce pro syntetizaci impedance

Z rešerše vyplynuly dva možné přístupy, jak digitálně syntetizovat impedanci použitelnou v oblasti aktivního tlumení vibrací. Nicméně bylo potřeba vybrat jen jeden z nich, analyzovat jej, realizovat a experimentálně odzkoušet. Volbu přístupu lze ještě dále rozdělit na dvě části. První je výběr topologie obvodu. Druhou pak výběr typu zařízení, které bude srdcem digitální části obvodu a bude provádět výpočty. *Univerzální obvodový simulátor* byl v kapitole 2.2 popsán s obvodem typu FPGA. Jistě je ale možné výpočet provádět i pomocí jiných typů obvodů, například dostatečně výkonným jednočipovým mikropočítačem. Proto jsem se rozhodl zvažovat více možností.

Nejprve jsem vybíral topologii obvodu pro další práci. Výhodou topologie *univerzálního obvodového simulátoru* se dvěma uzly popsaného v kapitole 2.2 je všestrannost. S jejím použitím by bylo možné vytvořit zařízení, u kterého by bylo možné měnit pomocí relé povahu vstupně-výstupních uzlů. V podstatě jedinou komplikací je skutečnost, že je třeba měnit povahu výstupu D/A převodníku, nicméně i to lze řešit relativně jednoduchým obvodem s operačními zesilovači. Tato koncepce rovněž umožňuje různou konfiguraci syntetizované impedance. V případě uvažovaných dvou vstupně-výstupních uzlů by tak bylo možné syntetizovat impedanci právě vzhledem k těmto dvěma uzlům nebo vůči jednomu vstupně-výstupnímu uzlu, přičemž druhý pól impedance by byl nulový potenciál obvodu. Na druhou stranu se realizace takového přístupu jeví rozhodně komplikovanější než u *obvodu syntetické admitance* popsaného v kapitole 2.3.

Druhý princip sice neumožňuje takovou všestrannost konfigurací jako první popisovaný. Naopak měří napětí mezi dvěma svorkami, na jehož základě určuje proud, a tak syntetizuje impedanční chování. Jediná možnost volby spočívá v hodnotě odporu  $R_C$ . V případě *obvodu syntetické admitance* postačuje jeden A/D převodník a jeden D/A převodník. U prvního přístupu, s proudovými konvejsory, jsou potřeba minimálně dva převodníky od každého typu, případně jeden dvoukanálový A/D resp. D/A převodník. Nicméně i tak každý kanál musí mít své obslužné obvody, jako např. obvod impedančního přizpůsobení. I přes relativní jednoduchost druhého přístupu byla experimentálně prokázána jeho plná funkčnost v přímé aplikaci aktivního tlumení vibrací. Proto jsem se rozhodl k praktické realizaci využít právě koncepci *obvodu syntetické admitance*. Myslím si, že pokud má být výsledek této práce vývojovým



stupněm k zařízení, které by mělo být schopno praktického nasazení, je jednoduchost návrhu velmi důležitá.

Po zvolení topologie obvodu jsem přistoupil k výběru vhodného prostředku, který bude zajišťovat digitální zpracování signálu. Bylo nutné zvolit vhodné, již existující, hardwarové řešení. Přístup, který byl použit Flemingem aj. (2000), tedy komplexní systém pro digitální zpracování signálu, který byl schopen implementovat potřebné spojité funkce na základě spolupráce s programem MATLAB, není pro tuto práci vhodný. Jistě by byl schopen zajistit spolehlivou funkci, nicméně výsledkem práce má být nejen laboratorní přístroj, ale hlavně by zvolené řešení mělo být předstupněm k zařízením schopným praktického nasazení. A proto je nutné, aby náklady na výrobu byly relativně nízké, což u hotových laboratorních řešení není pravidlem. Zejména je ale nutné, aby případný konstruktér navazujícího zařízení měl nad celým systémem co největší kontrolu. Tedy aby bylo možné ovlivnit veškeré operace se signály jak v analogové části tak i v digitální. A to může být u hotových řešení digitálního zpracování signálu problematické.

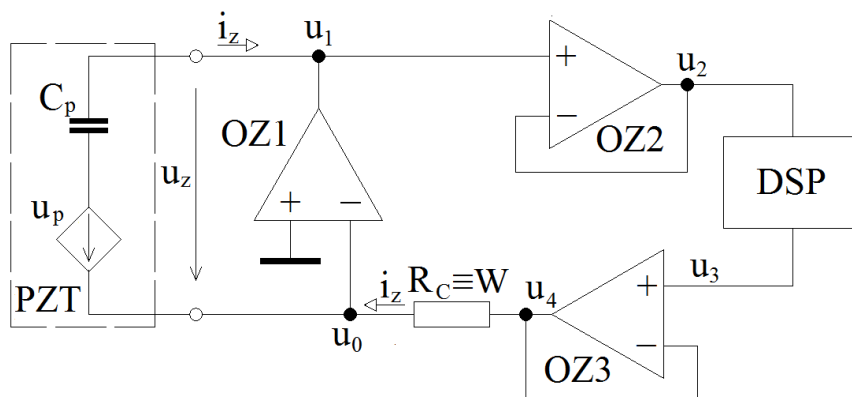
Co se týče typu obvodu pro digitální zpracování signálu, zvažoval jsem dvě základní možnosti. První bylo použití systému, jehož srdcem by byl obvod typu FPGA. Druhou pak nasazení prostředku, který by byl založen na dostatečně výkonném jednočipovém mikropočítači. Bylo vybráno řešení s mikropočítačem s architekturou ARM, který obsahuje aritmetickou jednotku umožňující výpočty s plovoucí desetinnou čárkou. Použití obvodu typu FPGA by jistě vedlo k vyšší rychlosti zpracování dat, je ale otázkou, zda by se tato přednost uplatnila při daném frekvenčním rozsahu do 5 kHz. Nabízela se dvě hardwarová řešení s vybraným obvodem ARM. Buď komerčně vyráběná vývojová deska nebo hardwarové řešení vyvinuté Nečáskem (2014), který se zabýval vývojem impedančního analyzátoru paralelně s touto prací. Navíc vyvíjený analyzátor má v budoucnu tvořit stejný tlumicí systém s obvodem syntetické impedance. Rozhodl jsem se použít upravené řešení kolegy Nečásky (2014), protože plně vyhovovalo potřebám práce. Navíc skutečnost, že v jednom systému v jedné laboratoři bude použito jedno hardwarové řešení, ulehčí práci při výzkumu a případné navazující konstrukci výsledných zařízení pro praktické nasazení.



### 3 Teoretická analýza a volba algoritmů

#### 3.1 Elektrická analýza

Nejprve jsem přistoupil k analýze zvolené koncepce. Zdroje, z kterých je převzata, poskytují spíše celkový základní popis a nevysvětlují podrobně funkci každé části obvodu. Při všech analýzách a výpočtech směřujících ke stanovení podmínek pro její reálnou implementaci je předpokládána linearita celého systému. Dále se předpokládá vstupní budící signál, který vzniká v piezočlenu a lze jej rozložit na součet harmonických signálů. Dílčí harmonické signály jsou v zadaném frekvenčním rozsahu, tedy od 50 Hz do 5 kHz. Rovněž se předpokládá pro účely analýzy základní funkce obvodu ideální chování operačních zesilovačů v daném pásmu.



Obr. 3.1 Elektrické schéma pro analýzu

Elektrické zapojení obvodu s vyznačením potřebných veličin je na obr. 3.1. Na místě vnějšího obvodu je náhradní elektrické schéma piezočlenu. Všechna napětí, která nejsou vyznačena šipkou, představují napětí daného bodu vůči nulovému potenciálu obvodu. Operační zesilovač OZ1 má připojen neinvertující vstup právě na nulové napětí. Invertující vstup pak k bodu s napětím  $u_0$ . Je-li tedy napětí  $u_0$  různé od nuly, OZ1 zasáhne pomocí výstupu tak, aby se dostalo zpět na nulu. Jeho funkci lze tak považovat za určitý regulátor udržující trvalou hodnotu  $u_0$  ve stavu tzv. virtuální nuly. Díky tomu napětí  $u_1$  odpovídá napětí  $u_z$ . Druhý operační zesilovač OZ2 plní pouze úlohu impedančního oddělovače, je zapojen jako napěťový sledovač, a tak má napětí  $u_2$  totožnou hodnotu s napětím  $u_1$ . Díky tomu lze zapsat vztah:

$$u_2 = u_z \quad (3.1)$$



Před popisem další části obvodu na obr. 3.1 bych nejprve uvedl, že jsem se rozhodl změnit oproti původnímu zdroji označení odporu  $R_C$  na  $W$ . V podstatě všechna náhradní elektrická schémata syntetizovaných impedancí obsahují činný odpor  $R$ . Mohlo by tedy dojít k záměně a výpočty by vypadaly nejasně. Blok DSP na základě napětí  $u_z$  určí funkcí  $f_{DSP}$  hodnotu  $u_3$  dle vztahu (3.2):

$$u_3 = f_{DSP}(u_z) \quad (3.2)$$

$$i_z = \frac{u_3}{W} \quad (3.3)$$

Napětí  $u_3$  je následně přeneseno pomocí OZ3 na napětí  $u_4$ . Vzhledem k tomu, že napětí  $u_0$  je udržováno na nulové hodnotě, lze pak velikost proudu  $i_z$  vyjádřit pomocí vztahu (3.3). Proud  $i_z$  teče z výstupu OZ3 přes odpor  $W$  a vnější obvod do výstupu OZ1. To vše díky uvažování nekonečných vstupních impedancí operačních zesilovačů. Zároveň je zřejmé, že tyto dva zesilovače (OZ1 a OZ3) musí mít stejné výstupní parametry, co se týče napětí a proudového rozsahu.

Až dosud byl popis v okamžitých hodnotách obvodových veličin. Vzhledem k tomu, že je uvažováno buzení obvodu harmonickým signálem, resp. jejich superpozicí, lze přejít k popisu pomocí komplexních veličin. Zároveň lze nahradit funkci  $f_{DSP}$  ve vzorci (3.2) pomocí lineárního napěťového přenosu  $F_U(j\omega)$  bloku DSP. Obvodové vztahy pak jsou ve tvaru:

$$\hat{U}_3 = F_U(j\omega) \cdot \hat{U}_z \quad (3.4)$$

$$\hat{I}_z = \frac{\hat{U}_3}{W} \quad (3.5)$$

$$\bar{Z} = \frac{\hat{U}_z}{\hat{I}_z} \quad (3.6)$$

$$\bar{Z} = \frac{W}{F_U(j\omega)} \quad (3.7)$$

Po dosazení vztahu (3.4) do vztahu (3.5) za napětí  $\hat{U}_3$  a následné substituci výsledku do vztahu (3.6) lze po úpravě získat nejdůležitější vzorec obvodu (3.7). Ten určuje závislost syntetizované impedance  $\bar{Z}$  na přenosu DSP bloku  $F_U(j\omega)$  a hlavně na velikosti odporu  $W$ . Jak bylo výše uvedeno, DSP blok vytváří průběh admitance, nicméně přenos  $F_U(j\omega)$  není přímo admitance. Jednak z důvodu rozměru jedná se o přenos z napětí na napětí, tedy bezrozměrný, zatímco rozměr admitance je 1 Siemens. Za druhé je průběh



absolutní hodnoty impedance  $\bar{Z}$  škálován velikostí odporu  $W$ , takže charakteristika přenosu  $F_U(j\omega)$  je v absolutní hodnotě  $W$  násobně menší než průběh admitance. Zdrojové dokumenty pro tuto koncepci od Fleminga aj. (2000) a od Fleminga (2004) neuvádějí explicitně souvislost podobnou vztahu (3.7). Ze vztahu (3.7) vyplývá možnost měnit rozsah absolutní hodnoty jen změnou velikosti  $W$ , přičemž rozsah absolutních hodnot přenosu  $F_U(j\omega)$  může být stále stejný. Rozhodl jsem se tedy rozdělit zadaný rozsah absolutních hodnot na čtyři intervaly podle dekád. Absolutní hodnota přenosu  $F_U(j\omega)$  bude maximálně rovna jedné. Nejnižší hodnota bude  $W_1 = 10 \Omega$  poté pro každou dekádu až do  $W_4 = 10 k\Omega$ . Ze vztahu (3.7) vyplývá, že absolutní hodnota syntetizované impedance  $\bar{Z}$  je nepřímo úměrná absolutní hodnotě  $F_U(j\omega)$ . Hodnotu  $W$  je tedy nutno volit tak, aby byla nižší než nejmenší amplituda syntetizované impedance. V principu je možný i opak, znamenalo by to ale, že absolutní hodnota  $F_U(j\omega)$  by musela být větší než jedna, což by mohlo vést k saturaci výstupu D/A převodníku a navazujících obvodů.

Kromě analýzy funkce samotné bylo nutné stanovit i podmínky pro realizaci. Tedy v případě elektrické části pro reálné operační zesilovače. Jedna z nich už byla zmíněna, že operační zesilovače OZ1 a OZ3 musí mít stejné výstupní rozsahy napětí i proudu. Samozřejmě všechny operační zesilovače musí mít odpovídající rozsah vstupního napětí. Z hlediska budících signálů je také důležitou vlastností rychlost přeběhu operačních zesilovačů. Při uvažovaném dílčím harmonickém napětí  $u(t)$  dle vztahu (3.8) je rychlost přeběhu největší v okamžicích průchodu nulou, ať již směrem nahoru nebo dolů, což se liší jen znaménkem. Derivaci  $u(t)$  zachycuje vztah (3.9). Její nejvyšší hodnota, která se poté periodicky opakuje, je pro  $t = 0$  s. Podle vztahu (3.10) lze po dosazení vypočítat minimální potřebnou rychlost přeběhu  $SR_{MIN} [V \cdot s^{-1}]$  pro danou frekvenci  $f [Hz]$  a amplitudu signálu  $U_M [V]$ . V případě této práce je mezní hodnota dána (3.11), pro  $f = 5 kHz$  a  $U_M = 10 V$ .

$$u(t) = U_M \sin(2\pi ft) \quad (3.8)$$

$$u'(t) = 2\pi f U_M \cos(2\pi ft) \quad (3.9)$$

$$SR_{MIN} = 2\pi f U_M \quad (3.10)$$

$$SR_{MIN} = 3,14 \cdot 10^5 V \cdot s^{-1} \quad (3.11)$$

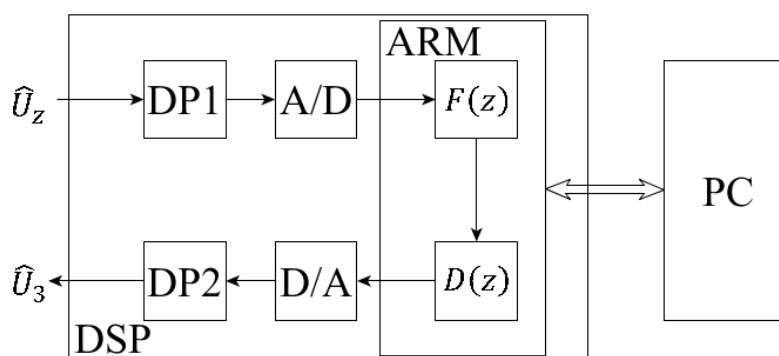




### 3.2 Analýza části zpracování signálu

Po analýze elektrické části jsem přistoupil k analýze části zpracování signálu, která byla na předchozích obrázcích zastoupena jen blokem označeným DSP s napěťovým frekvenčním přenosem  $F_U(j\omega)$ . Zahrnuje mikropočítač provádějící výpočet a zároveň i veškeré obvody spojené s A/D a D/A převody i analogovou část, která je nutná pro správnou funkci, především filtry. Cílem analýzy je popsat, jak všechny prvky části DSP ovlivňují parametry procházejícího signálu, a hlavně jak se projeví na frekvenčních průbězích výsledné syntetizované impedance  $\bar{Z}$ . Předpokladem je stejně jako v předchozím případě linearita všech součástí, digitální, převodní i analogové. Chyby A/D i D/A převodníku do analýzy nebudou zahrnuty.

Blokové schéma celé části pro zpracování signálu je na obr. 3.2. Srdce zpracování, tedy mikropočítač ARM, implementuje diskrétní přenosovou funkci  $F(z)$  dle obecného vzorce (3.12).  $N$  je řád jmenovatele  $M$  čitatele, přičemž hodnoty  $a_i$  a  $b_i$  jsou koeficienty jmenovatele resp. čitatele. Jedná se v podstatě o přenos filtru s nekonečnou odezvou označovaný zkratkou IIR. Vzhledem k tomu, že výpočet odezvy na základě implementace  $F(z)$  zabere určitý čas, dochází ke zpoždění signálu obecně o  $d$  vzorků. Zpoždění lze popsat přenosem  $D(z)$  dle vztahu (3.13). Prakticky bude zpoždění jeden vzorek, jakékoliv vyšší by vedlo k nefunkčnosti zařízení. Přenos korigovaný zpožděním  $F_D$  pak odpovídá vztahu (3.14).



Obr. 3.2 Blokové schéma části pro zpracování signálu

$$F(z) = \frac{b_0 + b_1 z^{-1} \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{1 + a_1 z^{-1} \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} \quad (3.12)$$

$$D(z) = z^{-d} \quad (3.13)$$

$$F_D(z) = F(z) \cdot z^{-1} \quad (3.14)$$



Mikropočítač je spojen s osobním počítačem PC, tak jsou předávány veškeré parametry výpočtu. Toto propojení plní jen obslužnou roli, tedy nemá přímý vliv na průběh zpracování signálu. Pouze je předáváno nastavení mikropočítače i celého obvodu. Proto PC není zahrnut do bloku DSP. Vstupem pro výpočet v mikropočítači je digitalizovaná hodnota napětí  $\hat{U}_z$ , která byla ještě před A/D převodem modifikována vstupním antialiasingovým filtrem DP1 s dolnoproustní charakteristikou. Jeho úkolem je zabránit signálům s vyšší frekvencí než polovinou vzorkovací frekvence, aby se dostaly do A/D převodníku a způsobily aliasing. Výsledek výpočtu je z mikropočítače předán D/A převodníku. Na jeho výstup navazuje druhý dolnoproustní filtr DP2 odstraňující složky vyšších frekvencí, které při diskretní realizaci samozřejmě vznikají. Výstupem je napětí  $\hat{U}_3$ , které je v navazující analogové části převedeno na proud  $\hat{I}_z$ .

Pro vyjádření závislosti napěťového přenosu  $F_U(j\omega)$  je frekvenční přenos prvního filtru  $F_{DP1}(j\omega)$  a druhého analogicky  $F_{DP2}(j\omega)$ . Nicméně nelze přímo použít diskretní přenos  $F_D(z)$  popisující realizovaný průběh. Je nutné použít jeho spojitý obraz, tedy  $F_D(j\omega)$ . Pak lze celkový  $F_U(j\omega)$  vyjádřit jako součin:

$$F_U(j\omega) = F_{DP1}(j\omega) \cdot F_D(j\omega) \cdot F_{DP2}(j\omega) \quad (3.15)$$

$$F_A(j\omega) = F_{DP1}(j\omega) \cdot F_{DP2}(j\omega) \quad (3.16)$$

$$F_U(j\omega) = F_A(j\omega) \cdot F_D(j\omega) \quad (3.17)$$

Vzhledem k předpokládané linearitě zpracování signálu lze pro zjednodušení spojit přenosy obou filtrů  $F_{DP1}(j\omega)$  a  $F_{DP2}(j\omega)$  do přenosu  $F_A(j\omega)$  dle (3.16). Vztah (3.15) poté přejde do podoby vzorce (3.17).

Dosud jsou všechny výpočty analytické. Nicméně vztah (3.17) nebude v analytické podobě potřeba k výpočtu. Jistě bude část predikce toho, jak je ovlivněna výsledná impedance  $\bar{Z}$ , vedena analyticky. Protože se však posuzuje frekvenční chování, musí být v určitém okamžiku z analytického popisu vypočteny vektory průběhu absolutní hodnoty a fáze vůči vektoru frekvence. Přenosy  $F_D(z)$  i  $F_A(j\omega)$  jsou komplexní. Nabízí se tedy možnost vypočítat vektory jejich amplitudy a fáze. S jejich pomocí se určí průběhy přenosu  $F_U(j\omega)$  a nakonec i impedance  $\bar{Z}$ . Vše dle pravidel o komplexních číslech v exponenciálním tvaru. Praktický vektorový výpočet bude probíhat dle následujícího popisu. Z přenosu  $F_D(z)$  se vypočte vektor hodnot fázové charakteristiky  $\vec{\varphi}_D$  a vektor pro absolutní hodnotu  $\vec{\psi}_D$ , obdobně pro přenos  $F_A(j\omega)$  vektory  $\vec{\varphi}_A$  a  $\vec{\psi}_A$ .



Složky vektorů budou příslušet frekvencím ve vektoru  $\vec{f}$ . Všechny vektory tedy musí mít délku  $K$ . Na základě pravidel pro počítání s komplexními čísly lze tedy vyjádřit vztahy pro charakteristiky přenosu  $F_U(j\omega)$  následovně:

$$\varphi_{U_k} = \varphi_{D_k} + \varphi_{A_k}; \quad k = 1 \dots K \quad (3.18)$$

$$\psi_{U_k} = \psi_{D_k} \cdot \psi_{A_k}; \quad k = 1 \dots K \quad (3.19)$$

Vzhledem k tomu, že ve vztahu (3.17) dochází k násobení přenosů, výpočet vektoru průběhu fáze  $\vec{\varphi}_U$  vede na součet jednotlivých složek dle vztahu (3.18). Pokud jde o vektor absolutní hodnoty  $\vec{\psi}_U$ , výsledkem je vektor násobků odpovídajících složek vektorů dle vztahu (3.19). Analogickou úpravou lze ze vztahu (3.7) přejít na vyjádření vektorů  $\vec{\varphi}_Z$  a  $\vec{\psi}_Z$  pro výslednou syntetizovanou impedanci  $\bar{Z}$ , ve vzorcích (3.20) a (3.21), odpor  $W$  je činný, s nulovou fází na všech frekvencích.

$$\varphi_{Z_k} = (-1) \cdot \varphi_{U_k}; \quad k = 1 \dots K \quad (3.20)$$

$$\psi_{Z_k} = \frac{W}{\psi_{U_k}}; \quad k = 1 \dots K \quad (3.21)$$

Z hlediska stanovení podmínek pro realizaci je nejdůležitější, aby implementovaný přenos filtru  $F(z)$  byl stabilní, kauzální, a tak realizovatelný. Z teorie digitálního zpracování signálů jsou tyto podmínky splněny, pokud všechny póly přenosu  $F(z)$  leží uprostřed jednotkové kružnice se středem v počátku komplexní roviny, žádný nesmí ležet ani na této kružnici. Pokud se rozloží jmenovatel  $F(z)$  na součin kořenových činitelů (3.22), musí pro všechny kořeny  $z_i$  platit nerovnost (3.23). To je tedy nejdůležitější nutná podmínka, aby absolutní hodnota všech kořenů jmenovatele přenosu  $F(z)$  byla menší než jedna. Připočtením přenosu  $D(z)$  dle (3.24) stabilita není ohrožena, protože zpožďovací přenos má pól v nule, tedy bezpečně uvnitř jednotkové kružnice. Filtry DP1 a DP2 musí dostatečně potlačit nežádoucí frekvence, zároveň ale nesmí příliš ovlivnit parametry syntetizované impedance, především fázi.

$$F(z) = \frac{b_0 + b_1 z^{-1} \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{(1 - z_1 z^{-1}) \cdot (1 - z_2 z^{-1}) \cdot \dots \cdot (1 - z_N z^{-1})} \quad (3.22)$$

$$|z_i| < 1; \quad \forall i \quad (3.23)$$

$$D(z) = z^{-1} = \frac{1}{z}; \quad z_1 = 0 \quad (3.24)$$

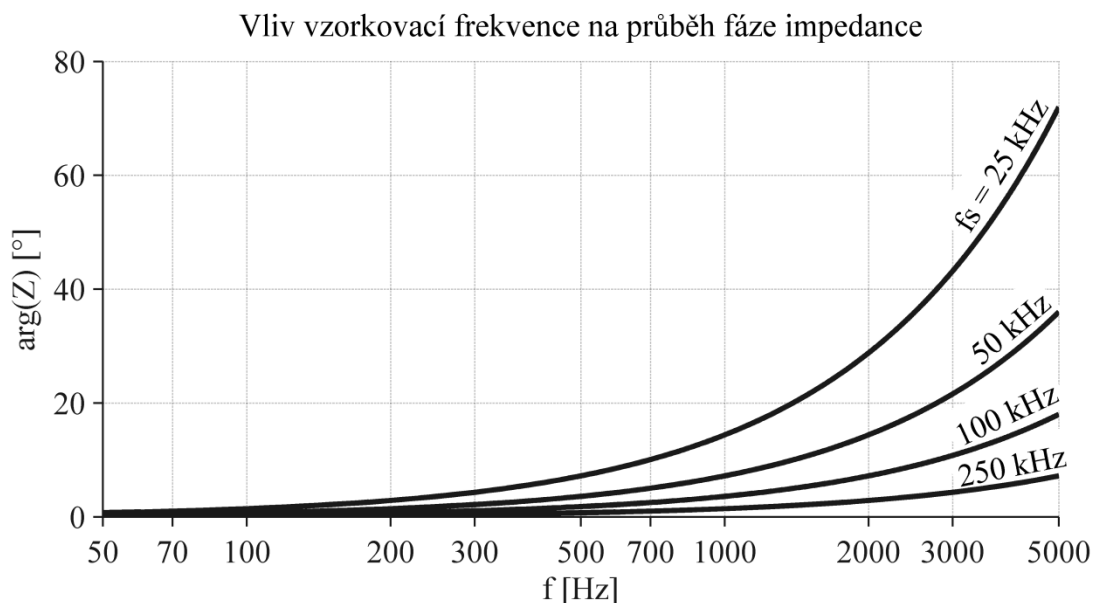


Poslední podmínka se týká volby vzorkovací frekvence  $f_s$ . Naprosté minimum vychází z Nyquistova teorému, aby  $f_s$  byla nejméně dvakrát vyšší než nejvyšší složka zpracovávaného signálu. Teoretické minimum je tedy hodnota 10 kHz. Z hlediska kvality syntetizované impedance však bude muset být vyšší. Přenos  $F(z)$  musí být doplněn na zpožděný  $F_D(z)$ . Pokud je  $F(z)$  v nejjednodušším možném tvaru dle (3.25) a tím (3.26), vede zpoždění na lineární fázovou charakteristiku. To vyplývá ze spojitého obrazu  $\bar{Z}$  ve vztahu (3.27), argument exponenciály narůstá lineárně s kruhovou frekvencí  $\omega$ , směrnice je převrácená hodnota  $f_s$ . Zároveň je vidět, že amplitudová charakteristika není dotčena, protože absolutní hodnota  $\bar{Z}$  je vždy rovna  $W$ . Graf. 3.1 zachycuje, jak se projeví toto zpoždění na průběhu fázové charakteristiky při různých hodnotách  $f_s$ . Byť jsou všechny závislosti fakticky lineární vůči frekvenci, vzhledem k logaritmickému měřítku mají odlišný tvar. Je vidět, že pro nízké hodnoty  $f_s$  je deformace fáze na vyšších frekvencích velká (až desítky stupňů). Této chyby fáze se v principu nelze zbavit, vždy bude přičtena k požadovanému průběhu, lze ji jen omezit použitím co nejvyšší možné hodnoty  $f_s$ .

$$F(z) = 1 \quad (3.25)$$

$$F_D(z) = z^{-1} \quad (3.26)$$

$$\bar{Z} = W \cdot e^{\frac{-j\omega}{f_s}} \quad (3.27)$$



**Graf. 3.1** Vliv vzorkovací frekvence na průběh fáze syntetizované impedance



### 3.3 Diskretizační metody

Všechny impedanční průběhy zadávané analyticky jsou vyjádřeny ve spojitě oblasti. Pro implementaci v mikropočítači je nutné přepočítat spojitou impedanci  $Z(s)$  na diskrétní přenos  $F(z)$ . Je tedy potřeba nejen provést výpočet podle vzorce (3.28), ale následně provést diskretizaci  $F(s)$ . Lze ji provést několika metodami, rozhodl jsem se tři z nich diskutovat. Dosud byly spojitě veličiny vyjádřeny jako závislé na kruhové frekvenci  $j\omega$ . V literatuře se popisuje diskretizace jako přechod z popisu pomocí komplexní proměnné  $s$  na diskrétní proměnnou  $z$ . Vzhledem k tomu, že pro periodické funkce je Fourierova transformace stejná jako Laplaceova, což je případ této práce, rozhodl jsem se přejít od popisu ve Fourierově oblasti na popis v Laplaceově oblasti. Prakticky to znamená jen jednoduchou symbolickou záměnu dle souvislosti (3.29). Navíc budou vzorce kratší a přehlednější, zároveň lze proměnnou  $s$  použít k symbolickým výpočtům i práci s přenosovými funkcemi v programu MATLAB, ve kterém jsou všechny výpočty vedeny.

$$F(s) = \frac{W}{Z(s)} \quad (3.28)$$

$$s = (j\omega) \quad (3.29)$$

První zvažovanou možností je dle popisu Forsytha (1991) bilineární aproximace, zvaná též Tustinova. Přechod od spojitěho přenosu spočívá v jednoduchém symbolickém nahrazení všech výskytů proměnné  $s$  dle (3.30). Případně pro práci se zápornými mocninami proměnné  $z$  lze postupovat pomocí vztahu (3.31).  $T$  představuje vzorkovací periodu, převrácenou hodnotu vzorkovací frekvence  $f_s$ , jak uvádí vztah (3.32). Známou výhodou této metody je, že všechny stabilní spojitě přenosy převádí na stabilní diskrétní přenosy. To znamená, že stabilní póly v  $s$ -oblasti (v levé komplexní polorovině) jsou převedeny na stabilní póly v  $z$ -rovině (uvnitř jednotkové kružnice).

$$s = \frac{2}{T} \cdot \frac{z - 1}{z + 1} \quad (3.30)$$

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad (3.31)$$

$$T = \frac{1}{f_s} \quad (3.32)$$



Druhou možností dle Forsytha (1991) je použití Adams-Bashforthovy metody. Postup je obdobný jako u předchozí. Spojitá proměnná  $s$  je symbolicky nahrazena pomocí vztahu (3.33) pro záporné mocniny proměnné  $z$ , nebo (3.34) pro kladné. Podle Forsytha (1991) je tato metoda vylepšením předchozí, tedy Tustinovy aproximace. Nicméně neuvádí, jak se Adams-Bashforthův přístup projeví na stabilitě diskrétního přenosu. Jak je vidět ze vztahu (3.33) i (3.34), vede navíc tato metoda na složitější popis v  $z$ -oblasti, protože substituce obsahuje proměnnou  $z$  v druhé mocnině. Bude-li se substituovat například za člen  $s^2$ , objeví se v diskrétním přenosu člen  $z^4$ . V první metodě vzorce (3.30) a (3.31) obsahují maximálně první mocninu, ať již kladnou nebo zápornou. Vysoký řád přenosů Adams-Bashforthovy metody tak může být podstatnou nevýhodou, protože prodlužuje výpočet odezvy filtru.

$$s = \frac{12}{T} \cdot \frac{1 - z^{-1}}{5 + 8z^{-1} - z^{-2}} \quad (3.33)$$

$$s = \frac{12}{T} \cdot \frac{z^2 - z}{5z^2 + 8z - 1} \quad (3.34)$$

Poslední zvažovanou možností bylo použití metody mapování pólů, publikovanou také Forsythem (1991), která zdá se, vykazuje velkou věrnost až do frekvencí blízkých  $f_s/2$ . Před popisem bych uvedl, že jsem oproti zdroji upravil symboliku tak, aby byla konzistentní s prací i obecnými konvencemi. V práci používám například koeficienty  $a_i$  pro jmenovatele a  $b_i$  pro čitatele, ve zdroji jsou použity obráceně patrně z důvodu, že publikace je věnována primárně návrhu regulátorů. Diskretizace se skládá ze dvou fází. Nejprve jsou ze spojitého přenosu  $F(s)$  dle (3.35) vypočteny póly  $s_i$ . Následně jsou všechny póly přeneseny, resp. mapovány na póly  $\alpha_i$ , do  $z$ -oblasti pomocí vztahu (3.36). Z kořenových činitelů je sestaven polynom jmenovatele  $p_D(z)$  diskrétního přenosu (3.37), následně přepočtený do tvaru (3.38). Tím byly mapovány všechny póly, zároveň je vidět, že řád jmenovatele spojitého přenosu je stejný jako diskrétního.

$$F(s) = \frac{h_L s^L + h_{L-1} s^{L-1} + \dots + h_1 s + h_0}{d_N s^N + d_{N-1} s^{N-1} + \dots + d_1 s + d_0} \quad (3.35)$$

$$\alpha_i = e^{s_i}; i = 1 \dots N \quad (3.36)$$

$$p_D(z) = (1 - \alpha_1 z^{-1})(1 - \alpha_2 z^{-1}) \dots (1 - \alpha_N z^{-1}) \quad (3.37)$$

$$p_D(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N} \quad (3.38)$$



Poté je nutno přistoupit k mapování nul přenosu  $F(s)$ . Dále uvedený popis byl proveden Forsythem (1991) jen na několika konkrétních případech. Musel jsem tedy podle jeho popisu provést odvození vztahů tak, aby bylo možné převádět obecnější přenosy. Odvození je pro přehlednost obsaženo v příloze B, zde uvádím pouze závěry a praktický postup výpočtu. Základem je výpočet dle vztahu (3.39). Výsledný vektor  $\vec{b}$  obsahuje koeficienty čitatele diskrétního přenosu ve tvaru dle (3.40). Délka  $M$  vektoru  $\vec{b}$ , tím i řád čitatele, je volena stejná jako u jmenovatele. Je ale v principu možné volit i delší čítel. Vektor  $\vec{a}$  ve tvaru (3.41) obsahuje již vypočtené koeficienty jmenovatele diskrétního přenosu dle (3.38).

$$\vec{b} = \mathbf{V}^{-1} \mathbf{H} \mathbf{D}^{-1} \mathbf{C} \vec{a} \quad (3.39)$$

$$\vec{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_M \end{bmatrix} \quad (3.40)$$

$$\vec{a} = \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} \quad (3.41)$$

$$\mathbf{D} = \begin{bmatrix} d_0 & 0 & 0 & \dots & 0 & 0 \\ d_1 & d_0 & 0 & \dots & 0 & 0 \\ \dots & d_1 & d_0 & \dots & 0 & 0 \\ d_N & \dots & d_1 & \dots & \dots & \dots \\ \dots & d_N & \dots & \dots & d_0 & 0 \\ 0 & \dots & d_N & \dots & d_1 & d_0 \end{bmatrix} \quad (3.42)$$

Dále je potřeba určit matice ve vztahu (3.39). Matice  $\mathbf{D}$  je čtvercová, její rozměr je  $M + 1$ , tedy o jednu větší než je řád  $M$  jmenovatele diskrétního přenosu. Její podobu zachycuje vztah (3.42). Jedná se o dolní trojúhelníkovou matici, která má všechny naddiagonální prvky nulové. Od hlavní diagonály směrem dolů pak každá diagonála přísluší jednomu koeficientu jmenovatele  $d_i$  původního spojitého přenosu. Vzhledem k tomu, že ve vztahu (3.39) se používá inverze matice  $\mathbf{D}^{-1}$ , je nutně nezbytné, aby matice  $\mathbf{D}$  byla regulární, musí mít nenulový determinant. Z matematické teorie to pro dolní trojúhelníkovou matici znamená, že musí mít všechny prvky na hlavní diagonále nenulové. V tomto případě je na diagonále jen jeden prvek  $d_0$ . Nutnou podmínkou pro použití tohoto postupu je právě nenulovost koeficientu  $d_0$ . Obdobně je tvořena další matice  $\mathbf{H}$ . Její strukturu zachycuje (3.43), prvky jsou tvořeny koeficienty  $h_i$  čitatele



původního spojitého přenosu. Je rovněž čtvercová a rozměr je roven  $M + 1$ , stejně jako u předchozí. Jak bylo výše uvedeno, řád diskrétního čitatele lze volit větší než  $N$ , nicméně zde jsem se omezil na volbu  $M = N$ . Pokud by byl čítec delší, vedlo by to na dodatečné zpoždění signálu a deformaci fáze charakteristiky.

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & 0 & \dots & 0 & 0 \\ h_1 & h_0 & 0 & \dots & 0 & 0 \\ \dots & h_1 & h_0 & \dots & 0 & 0 \\ h_L & \dots & h_1 & \dots & \dots & \dots \\ \dots & h_L & \dots & \dots & h_0 & 0 \\ 0 & \dots & h_L & \dots & h_1 & h_0 \end{bmatrix} \quad (3.43)$$

$$\mathbf{V}_{ij} = \mathbf{C}_{ij} = \frac{[-T \cdot (j - 1)]^{(i-1)}}{(i - 1)!} \quad (3.44)$$

Zbylé dvě matice  $\mathbf{C}$  a  $\mathbf{V}$  ze vztahu (3.39) jsou konstantní pro jednu vzorkovací periodu  $T$ . Jejich prvky jsou stejné a nejsou závislé na hodnotách koeficientů přenosů. Mění se jen rozměry  $\mathbf{C}$  a  $\mathbf{V}$ . Obecný vztah pro jednotlivé prvky  $\mathbf{V}_{ij}$  a  $\mathbf{C}_{ij}$  zachycuje vzorec (3.44), kde  $i$  představuje index řádku a  $j$  index sloupce. Matice  $\mathbf{V}$  je čtvercová a její rozměr je  $M + 1$ . Matice  $\mathbf{C}$  má  $M + 1$  řádků a  $N + 1$  sloupců. Při volbě  $M = N$  jsou tedy matice  $\mathbf{C}$  a  $\mathbf{V}$  stejné. Po výpočtu je z vektorů  $\vec{a}$  a  $\vec{b}$  složen přenos  $F(z)$ :

$$F(z) = \frac{b_0 + b_1 z^{-1} \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{1 + a_1 z^{-1} \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} \quad (3.45)$$

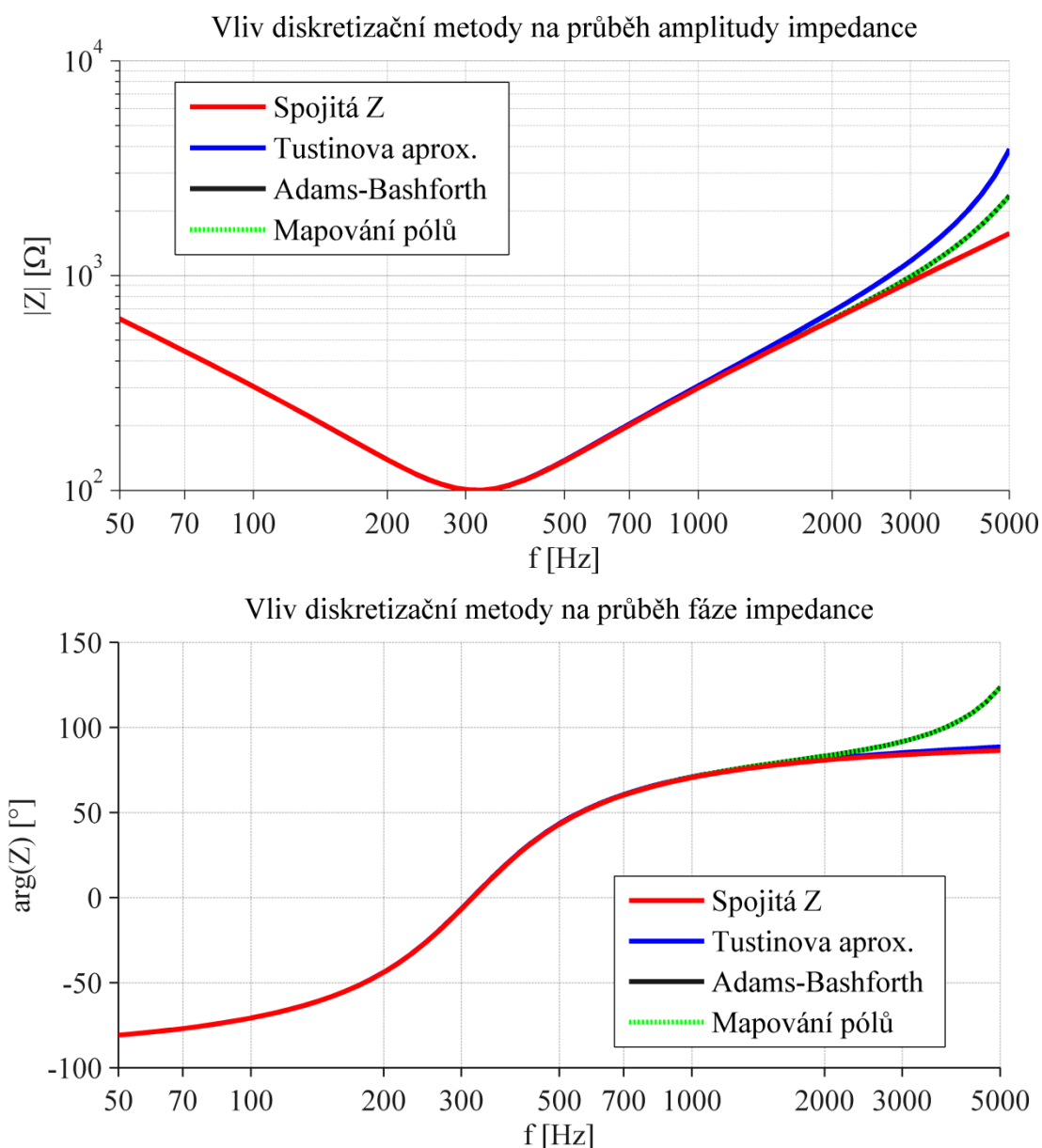
### 3.4 Diskretizace základních impedancí

Pro porovnání uvažovaných diskretizačních metod jsem vytvořil skript v MATLABu, ve kterém lze provést diskretizaci na libovolné impedanci zadané vzorcem. Pro první dvě metody, Tustinovu a Adams-Bashforthovu, byl použit symbolický výpočet příslušných substitucí. Metoda mapování pólů byla naprogramována tak, aby mohla pracovat co nejobecněji dle výše uvedeného popisu. Porovnání jsem provedl na průbězích základních impedancí popsanych v kapitole 2.1. Do přenosů nebyl započten vliv dodatečného zpoždění výpočtu  $D(z)$ , protože chyba fáze jím způsobená vzniká reálným výpočtem a je stejná pro všechny metody. Pro vyšší frekvence je chyba fáze díky  $D(z)$  tak velká, že by nebylo možné s jistotou porovnat vliv metod. Z porovnání vyplynulo, že pro vzorkovací frekvence nad 50 kHz nebyl mezi diskretizačními metodami prakticky žádný rozdíl v kvalitě výsledku. Průběhy nebyly odchýleny od originálního spojitého. Pro přenosy prvního řádu vedla Tustinova metoda





dokonce na stejné přenosy jako mapování pólů. Drobné odchylky v průbězích se objevily až při nižších vzorkovacích frekvencích. I na těch ale Adams-Bashforthova metoda vykazovala stejné výsledky jako mapování pólů. Nelze ani hovořit o nějaké systematické odchylce. Zatímco u jednoho typu impedance vykazovala lepší výsledky Tustinova metoda, jinde zbylé dvě. Pro ilustraci graf. 3.2 zachycuje průběhy impedance pro sériový RLC obvod (viz obr. 2.1, f), při  $f_s = 12,5 \text{ kHz}$ . V tomto případě vykazuje Tustinova metoda lepší chování ve fázi a naopak horší v amplitudě. Zbylé dvě metody naopak vedou na menší odchylku v amplitudě ale větší ve fázi. Oba průběhy se překrývají.



**Graf. 3.2** Vliv diskretizační metody na průběh amplitudy a fáze, při  $f_s = 12,5 \text{ kHz}$  a hodnotách  $W = 100 \text{ } \Omega$ ,  $R = 100 \text{ } \Omega$ ,  $L = 0,05 \text{ H}$   $C = 5 \text{ } \mu\text{F}$



Na základě srovnání jsem se rozhodl pro další práci používat Tustinovu metodu. Jednak proto, že kvůli omezení chyby fázové charakteristiky popsané v kapitole 3.2 musí být vzorkovací frekvence co nejvyšší,  $f_s = 50 \text{ kHz}$  je v podstatě použitelné minimum. Díky tomu vedou všechny metody na prakticky stejné výsledky. Dalším důvodem pro Tustinovu metodu je její jednoduchost, nezvyšuje řád systému a k jejímu výpočtu stačí prosté dosazení. Posledním důvodem je, že při jejím použití je stabilní spojitý přenos převeden na stabilní diskretní přenos. Zbylé dvě metody by v budoucnu přicházely v úvahu použít pouze ve zvláštních případech. Například pokud by z nějakého důvodu bylo nutné výrazně snížit vzorkovací frekvenci, přičemž by nevalila chyba fáze, a Tustinova metoda by pro dané použití nevyhovovala. Po výběru vhodné metody jsem vyřešil diskretizaci základních impedancí obecně tak, aby bylo možné v budoucnu jen dosadit hodnoty obvodových prvků do diskretního přenosu, výsledky jsou v příloze C. Odpadne tak nutnost symbolického výpočtu, který v případě laboratorního použití nevádí, ale budoucí konstrukci praktického zařízení by komplikoval.

Rovněž bylo nutné stanovit podmínky realizace jednotlivých impedancí, zvláště proto, že je nutné syntetizovat chování součástek se zápornými hodnotami, především kapacitou. Omezující podmínkou je stabilita přenosu  $F(z)$  implementovaného mikropočítačem. O stabilitě je možné rozhodnout ve spojitě oblasti a díky vlastnostem Tustinovy aproximace bude diskretizovaný přenos také stabilní. Je také výhodné stanovit omezující podmínky přímo pro tvar impedance  $Z(s)$ . Vzhledem k inverzi nul a pólů dle vztahu (3.46), tak omezení pro póly přenosu  $F(s)$  přejde na nuly impedance  $Z(s)$ ,  $W$  je reálné číslo, nemění rozložení kořenů. Všechny nuly  $n_i$  impedance  $Z(s)$  musí ležet v levé komplexní polorovině. Jako příklad je uvedena impedance sériového RC obvodu dle (3.47). Nulu lze určit dle vzorce (3.48) a výsledná podmínka má tvar (3.49). Příloha C obsahuje kromě diskretizovaných přenosů rovněž obdobné podmínky pro všechny základní impedance.

$$F(s) = \frac{W}{Z(s)} \quad (3.46)$$

$$Z(s) = \frac{sRC + 1}{sC} \quad (3.47)$$

$$n_1 = -\frac{1}{RC} \quad (3.48)$$

$$RC > 0 \quad (3.49)$$



### 3.5 Návrh filtru z tabulky hodnot

Dle zadání je potřeba syntetizovat i impedance, které nejsou popsány analyticky a jsou zadány tabulkou hodnot průběhů fáze a amplitudy. Zvažoval jsem použití konvenčních metod návrhu filtrů používaných v oblasti zpracování signálů. Většina z nich je však vytvořena tak, aby pracovaly se základními typy filtrů jako dolní propust, apod. V případě této práce však průběhy mohou být zcela obecné. Konvenční metody jsou také zpravidla navrženy hlavně s ohledem na průběh amplitudové charakteristiky. V určitých případech je jen požadován lineární průběh fázové charakteristiky nebo nejmenší možné zpoždění. V bočníkovém tlumení je ale správný průběh fáze velmi důležitý. Potřeboval jsem tedy metodu, která zohlední nejen průběh amplitudy, ale dostatečně i tvar fázové charakteristiky. Reálnou možností bylo použití integrované funkce MATLABu pro návrh na základě kritéria nejmenších čtverců. Ta ovšem má zásadní nevýhodu, že při výpočtu není uvažována realizovatelnost navrženého filtru. Výsledkem tedy může být filtr, který není stabilní nebo kauzální, a tudíž nerealizovatelný. Rozhodl jsem se proto řešit návrh filtru pomocí numerické optimalizace.

Jako první krok algoritmu jsou průběhy amplitudy a fáze impedance převedeny na průběhy filtru dle vztahů (3.50) a (3.51). Z vektoru průběhu fáze impedance  $\vec{\varphi}_Z$  je vypočten požadovaný vektor fáze filtru  $\vec{\varphi}_R$ . Obdobně je přepočten vektor absolutní hodnoty impedance  $\vec{\psi}_Z$  na požadovaný vektor filtru  $\vec{\psi}_R$ . Všechny vektory mají délku  $K$ , která je volena tak, aby byly všechny průběhy zachyceny s dostatečným frekvenčním rozlišením. Vektory  $\vec{\varphi}_R$  a  $\vec{\psi}_R$  jsou vzaty jako zadání optimalizační úlohy (viz dále). V případě potřeby jsou vhodně doplněny na frekvenční rozsah až do poloviny vzorkovací frekvence. Výsledkem úlohy jsou koeficienty  $a_i$  a  $b_i$  přenosu digitálního IIR filtru dle vztahu (3.52). Rozhodl jsem se formulovat úlohu i pro možnost FIR filtru, to znamená, že jmenovatel přenosu  $F(z)$  je roven jedné. Odvození je vedeno pro obecnější IIR a FIR varianta je jejím zjednodušením.

$$\varphi_{R_k} = (-1) \cdot \varphi_{Z_k}; \quad k = 1 \dots K \quad (3.50)$$

$$\psi_{R_k} = \frac{W}{\psi_{Z_k}}; \quad k = 1 \dots K \quad (3.51)$$



$$F(z) = \frac{b_0 + b_1 z^{-1} \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{1 + a_1 z^{-1} \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} \quad (3.52)$$

Pro kritériální funkci je nutné vypočítat v každém kroku minimalizace odezvu pro aktuální hodnoty koeficientů a určit vektory průběhů fáze a amplitudy. Dle vztahu (3.53) je vypočten řádkový komplexní vektor  $\vec{g}$  aktuálního frekvenčního průběhu. V případě FIR filtru vzorec neobsahuje sumu ve jmenovateli. Vektor  $\vec{f}_D$  obsahuje digitální frekvence. Jeho složky lze vypočítat dle vztahu (3.54).

$$g_k = \frac{\sum_{m=0}^M b_m \cdot \exp(-j \cdot m \cdot f_{Dk})}{1 + \sum_{n=1}^N a_n \cdot \exp(-j \cdot n \cdot f_{Dk})}; k = 1 \dots K \quad (3.53)$$

$$f_{D_{k+1}} = \frac{k}{K} \pi; k = 0 \dots K - 1 \quad (3.54)$$

$$\varphi_k = \ar g(g_k); \psi_k = |g_k|; k = 1 \dots K \quad (3.55)$$

Následně je vypočten řádkový vektor fáze  $\vec{\varphi}$  a amplitudy  $\vec{\psi}$  dle vztahu (3.55). Aktuální hodnoty jsou poté odečteny od zadaných dle vzorce (3.56). Vznikají tak dva vektory odchylek,  $\overrightarrow{\Delta\varphi}$  pro fázi a  $\overrightarrow{\Delta\psi}$  pro amplitudu. Z nich lze formulovat kritériální funkci  $f_{crit}$  dle (3.57), kde  $\Phi$  a  $\Psi$  jsou diagonální čtvercové matice váhových koeficientů o rozměru  $K$ . S jejich pomocí lze jednak normalizovat číselné hodnoty, aby odchylky v jednotlivých bodech měly stejnou váhu. Lze také zdůraznit místa charakteristiky, kde je potřeba aproximovat průběh přesněji.

$$\Delta\varphi_k = \varphi_{R_k} - \varphi_k; \Delta\psi_k = \psi_{R_k} - \psi_k; k = 1 \dots K \quad (3.56)$$

$$f_{crit} = \overrightarrow{\Delta\varphi} \cdot \Phi \cdot \overrightarrow{\Delta\varphi}^T + \overrightarrow{\Delta\psi} \cdot \Psi \cdot \overrightarrow{\Delta\psi}^T \quad (3.57)$$

Nalezením minima funkce  $f_{crit}$  lze vypočítat optimální filtr ve smyslu nejmenších čtverců odchylek od zadání. V případě FIR filtru je formulace algoritmu kompletní. Pro nalezení koeficientů IIR filtru je však nutné doplnit další kritériální funkci  $c(a_1, \dots, a_N)$ , která bude vypovídat o stabilitě filtru v každém kroku. Jejím vstupem je vektor aktuálních hodnot koeficientů jmenovatele, z nich jsou vypočteny absolutní hodnoty pólů. Pokud je filtr stabilní, výsledkem funkce je záporné číslo, při nestabilitě kladné. Nulový výsledek odpovídá mezi stability. Hledání minima s takovýmto omezením je v matematické teorii nazýváno úlohou nelineárního programování.

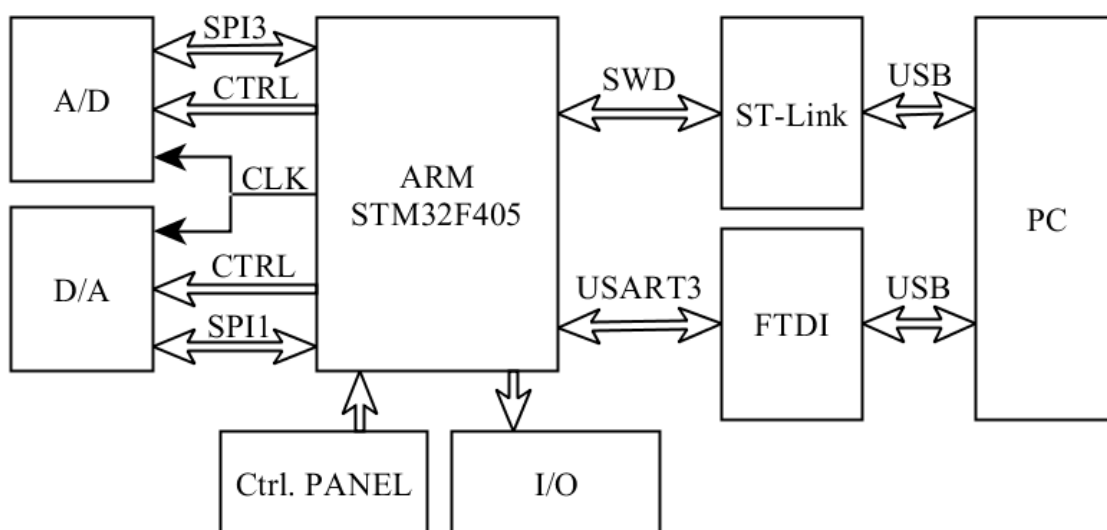


## 4 Hardwarové řešení

### 4.1 Digitální část

K praktické realizaci digitální části bylo vybráno hotové hardwarové řešení vyvinuté kolegou v rámci současně vznikající diplomové práce, která se zabývala jinou částí systému tlumení vibrací. Důvody pro tuto volbu byly shrnuty v kapitole 2.4. Nicméně nebylo možné aplikovat kompletní hardware v nezměněné podobě. Obsahoval několik prvků, které nebyly pro syntézu impedance potřeba, a zbytečně by zvyšovaly náklady na stavbu zařízení. Uvádím zde popis jen základních parametrů a hlavně rozdíly a úpravy, které byly provedeny. Podrobný popis hardwarového řešení zachycuje kolega Nečásek (2014) ve své paralelní práci.

Blokové schéma základní komunikace mikropočítače ARM je na obr. 4.1. Zachycuje propojení principiálně z hlediska funkce. Podrobné schéma zapojení obsahuje příloha D. Firmware je do mikropočítače zaváděn z počítače pomocí zařízení ST-Link protokolem SWD (Serial Wire Debug). S jeho pomocí lze také firmware ladit. Druhé propojení s PC je přes obvod FTDI převodníku. Ten umožňuje předávat data ze sériového rozhraní typu USART po USB do počítače. FTDI pomocí ovladače v PC poté vytvoří virtuální COM port. Díky tomu je možné pracovat s tímto propojením jednoduše jako se standardním sériovým portem. Pomocí něj je pak předáváno nastavení a parametry pro syntetizaci impedance. PC tak představuje nadřazený řídicí systém.



Obr. 4.1 Blokové schéma hlavních funkcí digitální části



S A/D a D/A převodníky je datová komunikace vedena pomocí sběrnic typu SPI. Čísla u jednotlivých sběrnic na obr. 4.1 ukazují, jaký byl použit komunikační modul mikropočítače ARM pro danou sběrnici. Obdobně pro sběrnici UART3 spojenou s FTDI obvodem. Kromě datové komunikace je mikropočítač ARM propojen s převodníky také několika řídicími signály, označenými zkratkou CTRL. Jedním z nejdůležitějších je hodinový signál CLK, který je společný pro oba převodníky. Uživatel ovládá zařízení pomocí řídicího panelu (Ctrl. PANEL) obsahujícího pět tlačítek. Digitální modul má dvanáct binárních výstupů, každý s indikační LED a optočlenem. Čtyři z nich jsem využil k přepínání výstupních rozsahů absolutní hodnoty syntetizované impedance, zbylé pak k signalizaci.

Výkres použitého plošného spoje pro modul digitální části je v netištěných přílohách. Použitá verze spoje se částečně liší od nejnovější vyvinuté, kterou použil kolega pro své výsledné zařízení. Podrobný popis změn je v práci Nečáska (2014). Rozdíly jsou však jen dílčí a netýkají se hlavních funkcí. V mé verzi oproti originálnímu návrhu nejsou osazeny některé součástky. Je to modul přídatné paměti SRAM a sériová paměť typu flash. Z důvodu úspor nebyly osazeny všechny optočleny, aby však byla zachována funkce indikačních LED, byly nahrazeny vhodným rezistorem. Pokud by se v budoucnu ukázalo jako potřebné dodat kterýkoliv z těchto prvků, je možné jej osadit.

## 4.2 Rozhraní analogové a digitální části

Mezi analogovou částí obvodu a digitální je rozhraní tvořené A/D a D/A převodníkem. A/D převodník měří napětí, které odpovídá napětí na svorkách syntetizované impedance. To je předáno do digitální části k výpočtu, jehož výsledkem je digitální slovo, které D/A převodník převádí na napětí odpovídající proudu syntetizované impedance. To vše je synchronně časováno z mikropočítače vzorkovací frekvencí  $f_s$ . Oba převodníky jsou spolu se všemi pomocnými obvody na jedné desce plošných spojů. Co se týká D/A převodníku, bylo použito hotové hardwarové řešení kolegy Nečáska (2014) s drobnou modifikací, spočívající v záměně typu samotného obvodu D/A převodníku. Původní 20bitový byl nahrazen 18bitovým. Jednak z důvodu úspory, ale především proto, že mnou použitý A/D převodník je rovněž 18bitový. Vzhledem k tomu, že oba typy D/A převodníků mají shodné pouzdro i zapojení vývodů, je motiv plošného spoje totožný s originálem. V tab. 4.1 jsou shrnuty základní údaje použitého D/A



převodníku. Hodnota rozlišení je podle (AD5781, 2010) skutečná, nemělo by docházet k jejímu snižování vlivem šumu. Dále kromě integrální nelinearity (INL) a diferenciální nelinearity (DNL) zachycuje tab. 4.1 maximální chybu v plném rozsahu (FSE) a maximální možnou vzorkovací frekvenci  $f_s$ .

**Tab. 4.1 Technické parametry použitého D/A převodníku, dle (AD5781, 2010)**

| AD5781    |                       |                       |              |
|-----------|-----------------------|-----------------------|--------------|
| rozlišení | 18 <i>b</i>           | max. $f_s$            | 1 <i>MHz</i> |
| INL       | $\pm 0,5$ <i>LSB</i>  | doba ustálení výstupu | 1 $\mu s$    |
| DNL       | $\pm 0,5$ <i>LSB</i>  | dolní mez výstupu     | −10 <i>V</i> |
| FSE       | $\pm 1,75$ <i>LSB</i> | horní mez výstupu     | 10 <i>V</i>  |

Pokud jde o A/D převodník, nebylo možné použít stejný typ jako v práci Nečáska (2014). Jedná se o velmi přesný sigma-delta převodník vhodný pro měření. S tím souvisí i skutečnost, že obsahuje relativně dlouhé digitální filtry, díky kterým je doba ustálení velká, a tak je skupinové zpoždění signálu značné. Pro syntézu impedance je ale naopak nutné co nejmenší zpoždění, jak bylo uvedeno v kapitole 3.2, zpoždění má vliv na průběh fázové charakteristiky. Proto jsem vybral rychlejší A/D převodník s postupnou aproximací, který neobsahuje žádné vnitřní filtry a nezpůsobuje tak velké zpoždění. Parametry použitého obvodu jsou v tab. 4.2. Ta kromě údajů o přesnosti zachycuje maximální hodnotu vzorkovací frekvence  $f_s$ , která je oproti D/A převodníku téměř poloviční a tvoří tak maximální použitelnou hranici. Rovněž je uveden odstup signálu od šumu (SNR) a dynamický rozsah. Obvodové zapojení převodníku včetně typů ostatních součástek bylo převzato z technické specifikace (AD7634, 2007). Upraveno bylo pouze připojení digitálního rozhraní, aby vyhovovalo použité koncepci. Meze měřitelného napětí v tab. 4.2 vycházejí ze zapojení a použitého napájecího napětí. Hodnoty uvedené v závorkách jsou maximální přípustné hodnoty na vstupu modulu A/D převodníku. Schémata zapojení obou převodníků jsou v příloze E. Kompletní výkres s plošným spojem je v netištěných přílohách.

**Tab. 4.2 Technické údaje použitého A/D převodníku, dle (AD7634, 2007)**

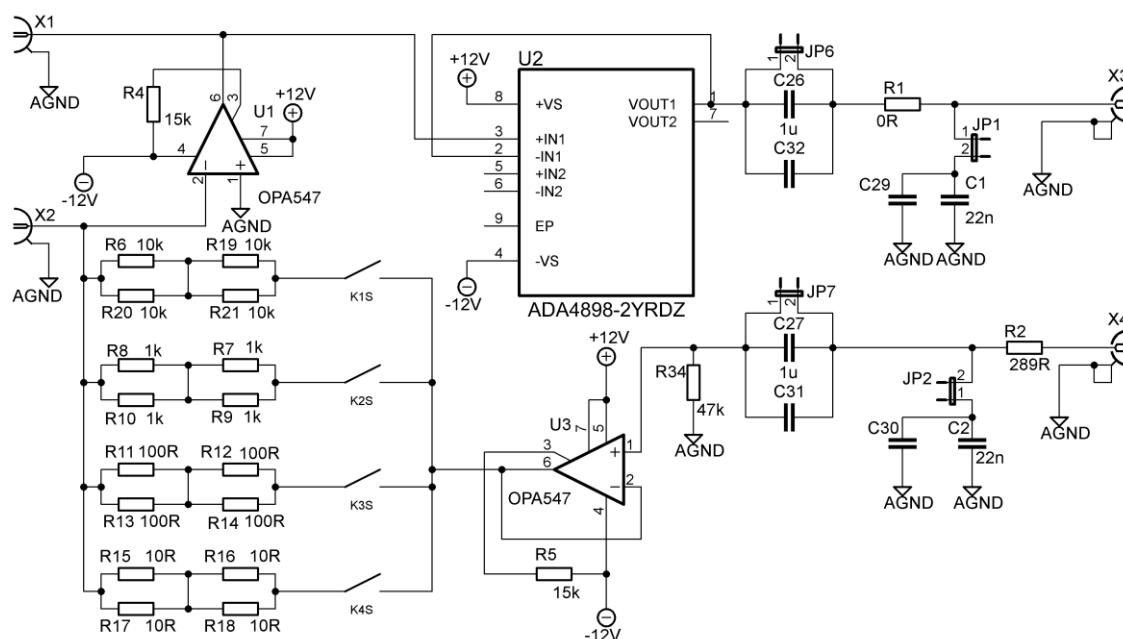
| AD7634                   |                      |                  |                              |
|--------------------------|----------------------|------------------|------------------------------|
| rozlišení                | 18 <i>b</i>          | SNR              | 101 <i>dB</i>                |
| INL (relativní přesnost) | $\pm 2,5$ <i>LSB</i> | max. $f_s$       | 570 <i>kHz</i>               |
| DNL                      | $\pm 2,5$ <i>LSB</i> | dolní mez vstupu | −10 <i>V</i> (−12 <i>V</i> ) |
| Dynam rozsah             | 102,5 <i>dB</i>      | horní mez vstupu | 10 <i>V</i> (12 <i>V</i> )   |



### 4.3 Analogová část

Zatímco v případě digitální části i modulu převodníků šlo alespoň z větší části použít hotové hardwarové řešení, pro analogovou část to nebylo možné. Pro ověření základní funkce obvodu jsem nejprve sestavil analogovou část na nepájivém poli. Po úspěšném vyzkoušení koncepce byla navržena analogová část jako samostatný plošný spoj. Podrobné schéma zapojení obsahuje příloha F. Na obr. 4.2 je jeho zjednodušená verze. Pro přehlednost obsahuje jen důležité prvky z hlediska průchodu signálu a základního principu. Podpůrné části jako blokovací kondenzátory, připojení napájení a ovládací část relé proto na obr. 4.2 nejsou.

Základem jsou dle zvolené koncepce na obr. 3.1 tři operační zesilovače (U1 až U3). Obvody U1 a U3, které ovlivňují výstup, jsou stejné výkonové zesilovače. Velikostí odporů R4 a R5 je určeno omezení jejich výstupních proudů. S použitou hodnotou je tak maximum výstupního proudu  $\pm 0,5\text{ A}$ . Maximum výstupního napětí je pak  $+10\text{ V}$  a  $-10,7\text{ V}$  proti nulovému potenciálu díky napájení  $\pm 12\text{ V}$  a vlastnostem zesilovačů. Rozsah absolutní hodnoty byl rozdělen do čtyř dekád, každá má své relé spínané z mikropočítače. Namísto jednoho rezistoru je hodnota složena ze čtyř z důvodu správného dimenzování ztrátového výkonu a hlavně jeho rozložení, aby nedošlo ke zkreslení hodnoty výrazným zahřátím. V případě nejvyššího rozsahu postačuje bezpečně jeden rezistor, nicméně z důvodu konzistence je návrh připraven na čtyři.

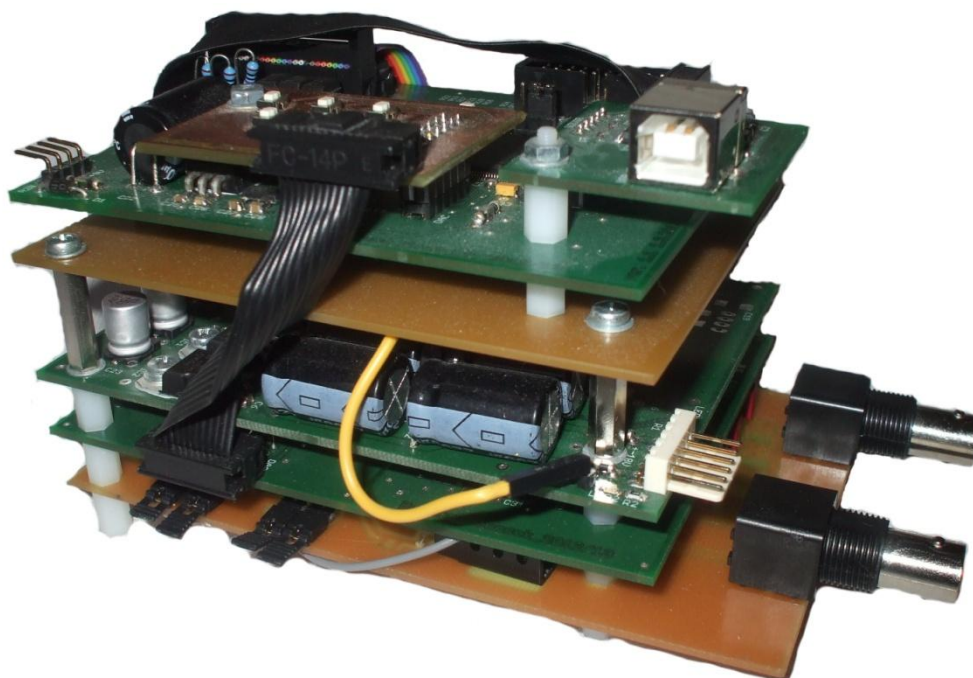


Obr. 4.2 Zjednodušené schéma analogového výstupního modulu





Poslední operační zesilovač U2 na obr. 4.2 je součástí obvodu dvojitého zesilovače. Druhá část však použita není, protože byl výstupní modul nejprve navržen s dalšími dvěma operačními zesilovači pro impedanční oddělení vstupů výkonových zesilovačů. Nicméně se nakonec v praxi ukázaly jako nadbytečné a spíše funkci komplikovaly, proto byly z funkce vyřazeny. Přehled všech změn na desce je součástí dokumentace v netištěných přílohách. Na výstup zesilovače U2 navazuje filtr před výstupem do A/D převodníku, složený z R1 a C1, resp. i C29. Kondenzátor s označením C29 je v zapojení čistě z praktických důvodů, aby bylo možné osadit více typů pouzder. Filtr lze vyřadit z funkce odstraněním propojky JP1. Stejný je princip vstupního filtru, který následuje za vstupem od D/A převodníku. Rezistor R34 zatěžuje signál a tím omezuje možné rušení kvůli vysoké vstupní impedanci zesilovače. Po vytažení propojek JP6 a JP7 prochází signál příslušnými kondenzátory. Díky tomu je možné zabránit průchodu stejnosměrného signálu v případě potřeby. Analogový modul je poslední část potřebného hardware. Byl integrován do koncepce ostatních částí, a tak tvoří jeden celek, který je zobrazen na obr. 4.3. Analogový modul je úplně dole pod deskou převodníků. Nad nimi je zdrojová část a úplně nahoře potom digitální modul s mikropočítačem.



**Obr. 4.3 Foto použitého hardware**



## 5 Programové vybavení

### 5.1 Firmware mikropočítače

Jednou částí programového vybavení je firmware mikropočítače ARM. Zajišťuje funkci celého hardware a přijímá nastavení z počítače, který představuje nadřazený řídicí systém. Tvorba a ladění kódu firmware probíhala v jazyce C ve vývojovém prostředí Atollic TrueSTUDIO for ARM Lite. Byly použity i hotové knihovny ve formě hlavičkových souborů umožňující například snadnější obsluhu a nastavení periférií pomocí maker a speciálních struktur. V této kapitole je popsána základní funkce a struktura firmware. Podrobnější popis poskytují komentáře přímo v kódu, který je součástí netištěných příloh.

Jako první zmiňuji z hlediska syntetizace impedance nejdůležitější funkci, kterou musí firmware zajistit. Jedná se o implementaci přenosu  $F(z)$  dle vzorce (5.1). Je tedy potřeba během jedné periody  $T$  vzorkovacího signálu nejprve načíst pomocí sériové sběrnice změřenou hodnotu z A/D převodníku a převést ji do formátu plovoucí desetinné čárky. Následně vypočítat odezvu, vrátit zpět do celočíselného formátu a poslat do D/A převodníku rovněž po sériové sběrnici. Pro samotný výpočet odezvy byla vybrána nejjednodušší možná metoda, tedy výpočet z diferenční rovnice. Z důvodu, že přenosy, které je potřeba implementovat, jsou nízkého řádu (do 10). V případě nutnosti realizovat filtry podstatně delší, by bylo vhodné zvážit jinou metodu, například kaskádní výpočet pomocí série přenosů prvního nebo druhého řádu. Prakticky je v kódu implementován vzorec (5.2). Celý kód firmware je nastaven tak, aby bylo možné použít filtry do devátého řádu, tedy maximálně pro  $N = 9$  a  $M = 9$ .

$$F(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} \quad (5.1)$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + \dots + b_M x[n-M] - a_1 y[n-1] - a_2 y[n-2] - \dots - a_N y[n-N] \quad (5.2)$$

Aby byla zajištěna správná synchronizace, je celý výše popsáný postup spouštěn asynchronním přerušením vnitřního časovače. Celková doba všech výše popsáných operací musí být menší než perioda vzorkování  $T$  s dostatečnou rezervou. Proto jsem v kódu, který implementuje tento periodický postup, použil jen nutné minimum operací. Aby bylo možné stanovit použitelné hodnoty vzorkovací periody  $T$  a tím i vzorkovací



frekvence  $f_s$ , provedl jsem orientační měření délky hlavních operací. Měření je orientační, protože byl osciloskopem měřen impulz na pomocném výstupu a většina údajů obsahuje navíc čekání na podmínku od externích obvodů, které nepracují synchronně s vykonáváním programu mikropočítače. Časy zahrnují i dobu přepnutí výstupu, ta je ale relativně malá. Kompletní impulz na výstupu trvá méně než 50 ns, což odpovídá v kódu nastavení logické jedna a v následujícím řádku ihned nulování. Výsledky jsou shrnuty v tab. 5.1. Doba výpočtu IIR pro daný řád zahrnuje převod změřené hodnoty z celočíselného formátu uint32 na typ s plovoucí čárkou float32, dále samotný výpočet diferenční rovnice, převod zpět z float32 do uint32 a také přesuny hodnot předchozích vstupů a výstupů v paměti. Doby celých přerušení zahrnují všechny výše popsané kroky pro daný řád filtru.

**Tab. 5.1 Orientační časy zpracování dat, měřeno osciloskopem Velleman APS 230**

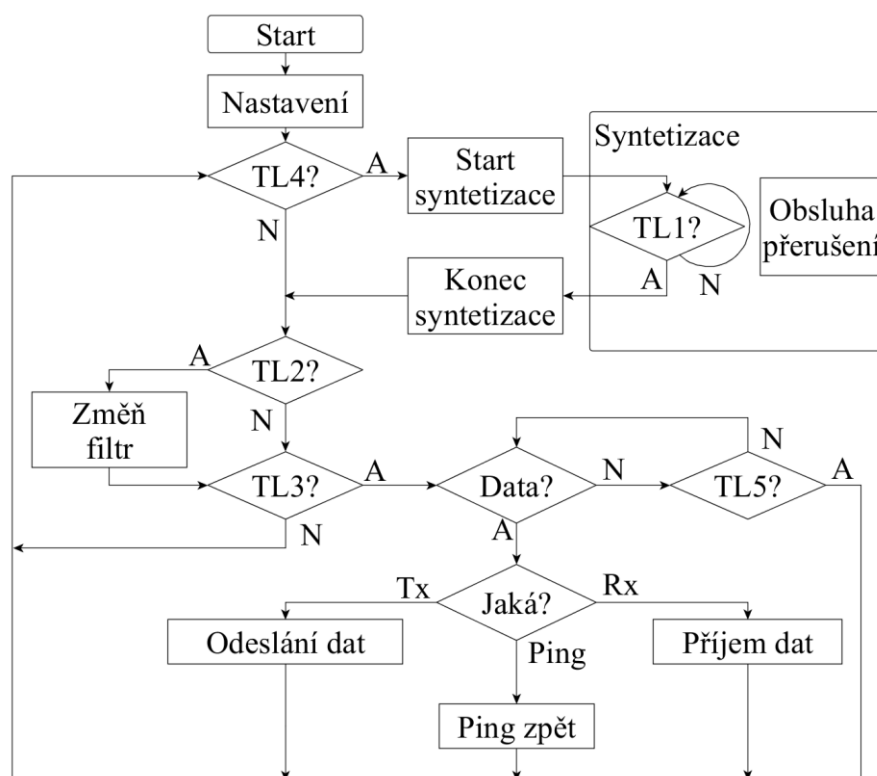
|   |               |
|---|---------------|
| Impulz na výstupu                         | < 50 ns       |
| A/D převod a vyčtení hodnoty z převodníku | 4,5 $\mu$ s   |
| Zápis hodnoty do D/A převodníku           | 2,23 $\mu$ s  |
| Výpočet IIR pro $M = N = 4$               | 2,12 $\mu$ s  |
| Výpočet IIR pro $M = N = 9$               | 4,08 $\mu$ s  |
| Celé přerušení pro $M = N = 4$            | 8,75 $\mu$ s  |
| Celé přerušení pro $M = N = 9$            | 10,92 $\mu$ s |

Co se týče změřených hodnot v tab. 5.1, nejdelší částí je doba A/D převodu a vyčtení jeho hodnoty, pravděpodobně z důvodu, že tento úsek obsahuje několik čekání na podmínku A/D převodníku a datového přenosu. Zápis do D/A převodníku oproti tomu trvá zhruba poloviční čas. Jako relativně dlouhá se mi jeví doba výpočtu diferenční rovnice. Použitý mikropočítač obsahuje jednotku pro počítání s plovoucí desetinnou čárkou. Pro  $M = N = 4$  se hlavní část výpočtu skládá z devíti operací násobení, čtyř sčítání, čtyř odčítání a osmi přesunů v paměti. Dle popisu instrukční sady v technické specifikaci (ARM, 2010) by tento úsek programu měl trvat v nejhorším případě 0,49  $\mu$ s. I tak je ale změřená doba výpočtu pětkrát delší. Abych vyloučil možnost chyby v programu, detailně jsem tuto část prozkoumal stejně tak i nastavení různých priorit asynchronního přerušení. Nicméně jsem žádnou chybu neobjevil a dobu výpočtu se mi zkrátit nepodařilo. Z toho důvodu jsem se rozhodl kód upravit tak, aby implementoval filtry IIR jen do čtvrtého řádu a bylo možné použít vzorkovací frekvenci alespoň 100 kHz. V případě potřeby implementovat delší filtr (až do devátého řádu) je nutné



odkomentovat příslušnou část kódu. Je nutné také stejným způsobem zmenšit vzorkovací frekvenci. Výše popsaná část firmware je zásadní pro syntézu impedance. Nicméně sama o sobě musí být doplněna o nastavovací procedury i jednoduché uživatelské rozhraní. Následuje popis celé struktury firmware, v níž je syntetizační funkce jen jednou částí.

Vývojový diagram na obr. 5.1 zachycuje hlavní kroky programu. Pro přehlednost nezachycuje detailně všechna jednotlivá volání funkcí, ukazuje především základní principiální strukturu firmware. Po startu programu je provedeno veškeré nastavení mikropočítače, periférií i výchozího přenosu filtru. Poté se program dostane do nekonečné smyčky. V ní je větven pomocí ovládacích tlačítek. Jejich čísla jsou dána fyzickým rozmístěním na panelu, ne významem nebo pořadím v běhu programu.



**Obr. 5.1 Vývojový diagram firmware mikropočítače**

Prvním testovaným tlačítkem je TL4. Pokud není stisknuto (N), program pokračuje na další. Pokud je TL4 stisknuto (A), odstartuje syntetizace. To znamená nulování souvisejících proměnných a povolení přerušení od příslušného časovače. Po této rutině se program dostane do nekonečné smyčky, ve které neustále testuje stisknutí tlačítka TL1. Zároveň je vyvoláváno přerušení s funkcí popsanou výše v této kapitole. Ukončit syntetizaci lze stisknutím tlačítka TL1, pak proběhne rutina ukončení syntetizace a program pokračuje testováním tlačítka TL2. Pokud není TL2 stisknuto, pokračuje



zjištěním stavu dalšího, pokud je, dochází ke změně filtru. Firmware je vytvořen tak, že umožňuje poslat do mikropočítače dvě různé konfigurace filtru, mezi kterými se přepíná právě TL2 spouštějícím příslušnou funkci.

Poslední tlačítko testované v hlavní smyčce je TL3. Tím se program uvede do režimu čekání na příchozí data z nadřazeného systému, tedy počítače. Hodnota prvního přijatého bytu rozhoduje mezi třemi reakcemi mikropočítače. Pokud odpovídá funkci ping, je stejný byte odeslán zpět, což slouží především pro odzkoušení komunikace. Další možností je režim příjmu dat. Při něm jsou přijaty hodnoty filtru ve formě dvou vektorů a také rozsah absolutní hodnoty. Vše je uloženo do aktuální konfigurace filtru. Poslední variantou je naopak odeslání dat o aktuální konfiguraci filtru zpět do počítače. Po provedení příslušné funkce se program vrací na začátek hlavní smyčky. V případě, že v režimu příjmu dat nepříjde žádný byte, lze režim opustit stiskem posledního tlačítka TL5, program se pak vrací rovněž na začátek hlavní smyčky.

Pro praktickou práci se zařízením jsem shrnul všechny základní ovládací a signalizační prvky v tab. 5.2. Obsahuje jak funkci již popsanych tlačítek TL1 až TL5, tak i tlačítko pro restart programu mikropočítače, které nebylo výše zmíněno. Tab. 5.2 zmiňuje i signalizační svítivé diody. LED1 až LED4 indikují aktuální rozsah absolutní hodnoty syntetizované impedance. Ostatní diody zobrazují zvolenou konfiguraci a režim, ve kterém mikropočítač pracuje. Co se týká probíhajícího režimu syntetizace, ten je signalizován LED10. Druhá dioda, která se jej týká, je uvedena v závorce, protože je buzena obdélníkovým signálem, jehož stav logické jedničky indikuje probíhající výpočet odezvy filtru. Lze tak osciloskopem měřit hodnotu vzorkovací frekvence a také dobu výpočtu filtru včetně všech převodů a datových přenosů. Diody neuvedené v tab. 5.2 zůstaly nevyužity.

**Tab. 5.2 Přehled naprogramovaných ovládacích a signalizačních prvků**

| Ovládání tlačítka    |                      | Signalizace         |                |
|----------------------|----------------------|---------------------|----------------|
|                      |                      | $W_1 = 10 \Omega$   | LED1           |
| Start syntetizace    | TL4                  | $W_2 = 100 \Omega$  | LED2           |
| Stop syntetizace     | TL1                  | $W_3 = 1 k\Omega$   | LED3           |
| Přepnutí konfigurace | TL2                  | $W_4 = 10 k\Omega$  | LED4           |
| Režim komunikace     | TL3                  | Konfigurace 1       | LED8           |
| Zrušení komunikace   | TL5                  | Konfigurace 2       | LED9           |
| Reset běhu programu  | Samostatné vedle LED | Syntetizace probíhá | LED10, (LED11) |
|                      |                      | Režim komunikace    | LED12          |



## 5.2 Program v počítači

V počítači byla veškerá tvorba obslužných programů a implementace algoritmů provedena v MATLABu pomocí skriptů. Tento přístup byl zvolen s ohledem na skutečnost, že zařízení má být používáno v laboratoři, ve které jsou měření koncipována stejným přístupem. Rovněž je možné použít symbolické výpočty. Uživatel mění parametry syntetizace a zadává hodnoty přímo v kódech jednotlivých skriptů. Přepínání variant výpočtu je realizováno zakomentováním resp. odkomentováním příslušných částí. V této kapitole popisují základní strukturu vytvořených kódů a jejich klíčové prvky. Podrobnější popis poskytují komentáře přímo ve skriptech, které jsou jak v netištěných přílohách, tak v příloze G. Co se týče značení vektorů, řecká písmena a symboly s indexy použité v teoretické části jsou vhodná pro zápis vzorců v textu, ale pro programování nejsou praktická. Proto mají vektory a jednotlivé veličiny ve skriptech většinou delší popisné názvy.

První vytvořený soubor nese název *PorovnaniDiskretizacnichMetod.m* a s jeho pomocí jsem provedl srovnání uvažovaných diskretizačních metod na základních typech impedancí. Uživatel nejprve zadá parametry, tj. vzorkovací frekvenci  $f_s$  a rozsah absolutní hodnoty  $W$ . Pro každý typ základní impedance je připraven příslušný blok skriptu. Uživatel jej odkomentuje a zadá hodnoty. Ostatní bloky musí být zakomentovány. V případě nutnosti diskretizovat jiný typ impedance je nutné vytvořit obdobný blok se zadáním hodnot a hlavně vzorcem pro spojitou impedanci. V další fázi programu je spojitá impedance  $Z(s)$  převedena na přenos filtru pro implementaci  $F(s)$  dle popisu v kapitole 3.2. Následně je  $F(s)$  diskretizován dle kapitoly 3.3 Adams-Bashforthovou a Tustinovou metodou pomocí symbolického výpočtu a integrované funkce. Poté je  $F(s)$  diskretizován pomocí metody mapování pólů, což obnáší několik kroků. Nejprve jsou mapovány póly jmenovatele, následně jsou vytvořeny všechny potřebné matice a vektory dle vztahů (3.41), (3.42), (3.43) a (3.44). Po výpočtu klíčové rovnice této metody je z jejího výsledku sestaven poslední diskretizovaný přenos. Jsou vypočteny frekvenční průběhy všech diskretizovaných impedancí a spolu s původní spojitou zobrazeny ve dvou grafech pro fázi a amplitudu. Popsaný skript slouží pouze k teoretické rozvaze, proto není nijak spojen s přenosem dat do mikropočítače.

Další skripty již vytvářejí výsledky, které je nutné přenést jako nastavení do mikropočítače. Manipulace s daty je následující. Daný skript vytvoří přenos filtru  $F(z)$ ,



který je nutno implementovat. Zároveň vznikají vektory koeficientů čitatele i jmenovatele, oba v datovém formátu kompatibilním s aritmetikou mikropočítače ARM. Rovněž je vygenerován index rozsahu absolutní hodnoty syntetizované impedance. Všechny tyto tři proměnné jsou skriptem pod specifickými názvy uloženy v pracovním prostoru MATLABu a nejsou ihned poslány, protože často je nutné měnit parametry, a tak spustit daný skript několikrát než je dosaženo relevantního výsledku. V případě, že uživatel chce zapsat vytvořené nastavení do mikropočítače, připojí jej k počítači, uvede do příslušného režimu a spustí skript s názvem *ZapisDatRS232.m*. Ten nejprve otestuje, zda jsou potřebné proměnné v pracovním prostoru. Pokud ano, otevře sériový port a pošle všechna data. Ve skriptu jsou ošetřeny chyby přenosu, především je zajištěno, aby vždy po skončení byl příslušný port uzavřen. V opačném případě by mohlo dojít k chybě, kvůli které by bylo nutné restartovat MATLAB. Pokud uživatel naopak potřebuje zjistit aktuální konfiguraci mikropočítače, uvede jej do příslušného režimu a poté spustí skript *CteniDatRS232.m*, který vyčte koeficienty filtru a index rozsahu absolutní hodnoty a vše zobrazí v prostředí MATLABu. Vytvořil jsem také dva velmi krátké skripty s názvy *UlozeniDatKOdeslani.m* a rovněž *NacteniDatKOdeslani.m*, s jejichž pomocí lze uložit proměnné z pracovního prostoru a znovu do něj načíst. Je nutné jen zadat vhodný název.

Prvním skriptem, který vytváří přenos  $F(z)$  k implementaci, je soubor s názvem *ZakladniImpedance.m*. Jeho struktura je podobná skriptu k porovnání diskretizačních metod. Uživatel rovněž vybere potřebný typ základní impedance, rozsah absolutní hodnoty  $W$ , vzorkovací frekvenci  $f_s$  a zadá hodnoty součástek. Nejprve je otestováno kritérium stability. Pokud hodnoty vyhovují, spojitá impedance  $Z(s)$  je přepočtena na přenos  $F(s)$ , který je následně diskretizován pomocí Tustinovy metody. Jsou vypočteny frekvenční průběhy ideální spojitě impedance a diskrétní verze, která zahrnuje jak zpoždění díky výpočtu, tak vliv filtrů v obvodu. Všechny průběhy jsou zobrazeny. Zároveň jsou vytvořeny vektory koeficientů čitatele a jmenovatele i index odpovídající rozsahu absolutní hodnoty. Tyto tři proměnné uložené v pracovním prostoru jsou odeslány do mikropočítače dle výše popsaného postupu.

Další skripty implementují návrh filtru  $F(z)$  na základě dat zadaných tabulkou hodnot dle popisu v kapitole 3.5. Jsou to dva soubory. *NavrhFiltruFIR.m* pro výpočet FIR filtru a analogicky *NavrhFiltruIIR.m*. Jejich základní struktura je totožná, mají



společnou i většinu kódu, proto jsou popsány zároveň s uvedením vzájemných odlišností. Nejprve uživatel nastaví základní parametry jako vzorkovací frekvenci  $f_S$ , rozsah absolutní hodnoty  $W$  a řád filtru. U verze IIR lze volit řád čitatele i jmenovatele. Následně jsou načtena vstupní data ze souboru, která musí být vždy přizpůsobena zadaným parametrům. Poté jsou nastaveny vlastnosti frekvenční charakteristiky, která je rozdělena na tři části. Nejdůležitější z nich je oblast zájmu. Její hranice odpovídají minimu a maximu charakteristiky načtené ze vstupního souboru. Tuto oblast doplňují intervaly na nižších frekvencích (od 0 Hz) a na vyšších (až do  $f_S/2$ ) tak, aby při výpočtu byly i tyto hodnoty uvažovány. Uživatel volí počet kroků pro lineární rozdělení každé z těchto oblastí. Zpravidla má nejvíce kroků oblast zájmu a zbylé dvě řádově méně. Po tomto úseku kódu následuje nastavení výchozích hodnot pro numerickou optimalizaci. Uživatel volí mezi uloženými parametry v souboru nebo náhodně navrženými. V případě IIR filtru je pro jmenovatele navržen vektor stabilního polynomu.

Po nastavení počátečního bodu optimalizace následuje vygenerování váhových matic  $\Psi$  a  $\Phi$ , respektive jejich diagonálních prvků. Tuto část nelze naprogramovat zcela obecně. Uživatel musí vždy koeficienty přizpůsobit tvaru vstupních dat a potřebám syntetizace. Naprogramoval jsem dvě základní možnosti. První je lineární klesající funkce pro frekvence v oblasti zájmu. Druhou variantou je pak použití standardizované okénkovací funkce. Ve skriptu potom následuje část pro interpolaci zadaných dat. Ta je velmi důležitá, protože vstupní charakteristiky mají zpravidla jiný frekvenční krok než je potřeba pro výpočet. Proto jsou hodnoty amplitudy i fáze interpolovány pomocí funkce *spline.m* na potřebné frekvenční rozlišení. Následuje jednoduché doplnění zadaných průběhu amplitudy i fáze tak, aby byly pokryty všechny frekvence od nulové až do hodnoty  $f_S/2$ . Průběhy impedancí jsou zpravidla zadány jen v relativně úzkém pásmu.

Následuje korekce zadaných hodnot. V případě potřeby je možné modifikovat zadané průběhy tak, aby obsahovaly kompenzaci zpoždění výpočtu i přídavných filtrů v obvodu. Jejich vliv je popsán v kapitole 3.2. Pro kompenzaci je původní vektor fáze  $\vec{\varphi}$  upraven dle (5.3), kde  $\vec{\varphi}_N$  představuje korigované zadání a  $\vec{\varphi}_A$  zahrnuje vliv přídavných filtrů i zpoždění výpočtu. Analogicky je pak vztah pro amplitudu ve tvaru (5.4).

$$\varphi_{Nk} = \varphi_k - \varphi_{Ak}; \quad k = 1 \dots K \quad (5.3)$$

$$\psi_{Nk} = \psi_k / \psi_{Ak}; \quad k = 1 \dots K \quad (5.4)$$





Po provedení výše popsaných kroků následuje samotné řešení optimalizační úlohy. V případě verze FIR je použita funkce MATLABu *fminsearch.m*, která nalezne minimum kritériální funkce *FIRcritFvek.m* obsahující výpočet kritéria dle kapitoly 3.5. Ve verzi IIR provádí minimalizaci kritéria ve skriptu *IIRcritFvek.m* funkce *fmincon.m* umožňující optimalizaci s mnoha podmínkami. V případě této práce je však použito jen nelineární omezení funkcí *IIRRootsConstrain.m*, která v případě nestabilního přenosu vrací kladné číslo a v případě stability záporné. Kritérium stability bylo oproti teorii striktnější. Za stabilní jsou považovány přenosy, jejichž všechny póly mají absolutní hodnotu menší než 0,999. Teoretickou hranicí je sice jednotková kružnice, ale vzhledem k tomu, že v reálném výpočtu dochází k zaokrouhlování a koeficienty přenosu jsou implementovány s určitou přesností, bylo kritérium zpřísněno.

Vzhledem k rozhraní použitých funkcí MATLABu bylo nutné umístit koeficienty čitatele i jmenovatele do jednoho vektoru  $\vec{x}$  ve tvaru (5.5),  $a_i$  jsou hodnoty jmenovatele a  $b_i$  čitatele. Ve variantě FIR jsou hledány jen koeficienty čitatele,  $\vec{x}$  má stejný tvar, jen neobsahuje koeficienty  $a_i$ . Po provedení výpočtu optimalizace je ve verzi IIR testována proměnná určující, zda je výsledek optimalizace správný nebo ne. Funkce *fmincon.m* vrací mimo jiné číselnou hodnotu, která je kladná v případě přípustného výsledku a záporná v případě výsledku neodpovídajícího omezení, v tomto případě pro nestabilní přenos. Pokud je výsledek špatný, skript se ukončí a vypíše chybové hlášení.

$$\vec{x} = [b_0 \quad \dots \quad b_M \quad a_1 \quad \dots \quad a_N] \quad (5.5)$$

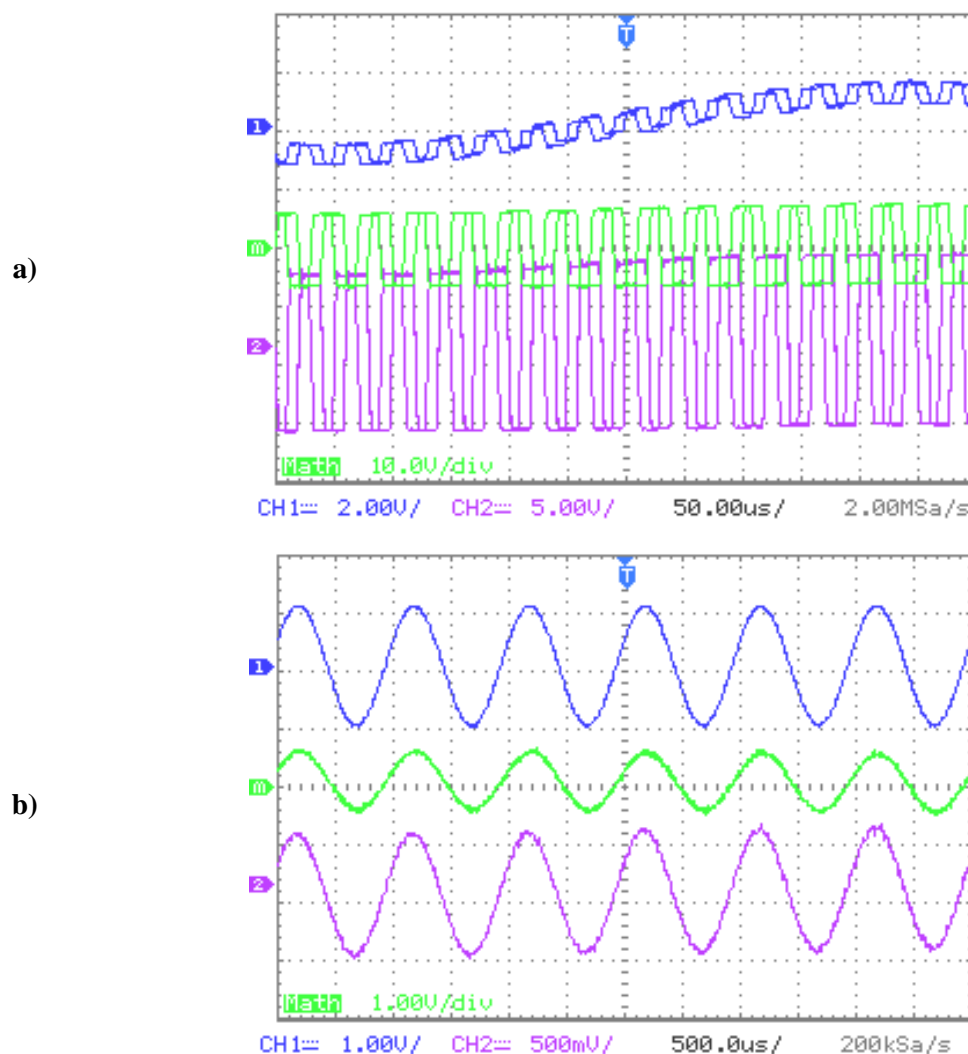
Po řešení úlohy následuje vytvoření proměnných pro odeslání do mikropočítače ARM obdobně jako ve výše uvedených případech. Poté jsou vypočteny charakteristiky navrženého filtru a zobrazeny spolu se zadáním ze vstupního souboru. Vzhledem k tomu, že řešení optimalizační úlohy je relativně komplikované a je nutné nastavit mnoho parametrů a veličin, zobrazení charakteristik je mnohem komplexnější než v předchozích skriptech. Umožňuje zobrazit jak impedanční průběhy, tak je možné zobrazovat i průběhy implementovaného přenosu  $F(z)$  vůči zadání v úzké oblasti zájmu nebo v celém frekvenčním rozsahu.



## 6 Experimentální ověření syntézy

### 6.1 Ověření základní funkce obvodu

Jako první bylo provedeno vyzkoušení zvolené koncepce, kdy analogová část byla sestavena na nepájivém poli. Ostatní části hardware byly již ve finální podobě. Byly změřeny průběhy některých impedancí. Nicméně výsledky zde neuvádím, účelem bylo především odzkoušet zapojení a celou koncepci. Syntetizované průběhy všech základních impedancí byly důkladně změřeny až s kompletním hardware a výsledky jsou diskutovány v samostatné kapitole 6.2. Při konfiguraci zařízení s částí na nepájivém poli bylo nalezeno vhodné nastavení filtrů DP1 a DP2 popsanych v kapitole 3.2, resp.

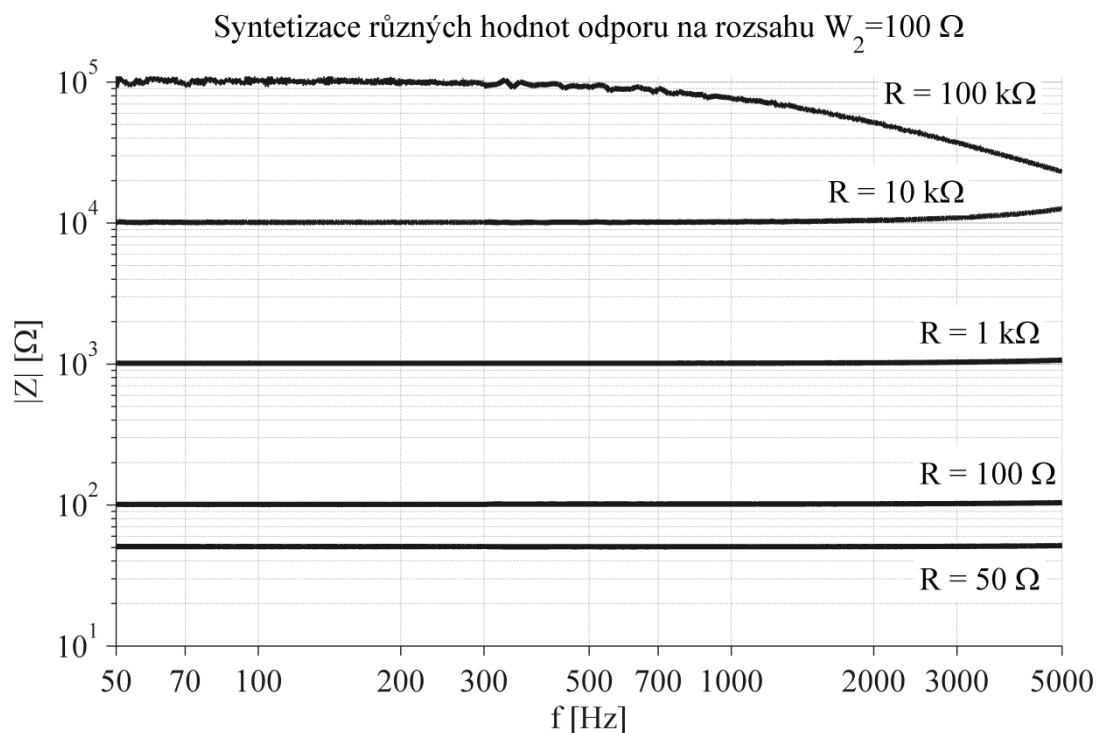


Obr. 6.1 Průběhy napětí a proudu syntetizované impedance, a) bez filtrace, b) s filtrací, měřeno osciloskopem Agilent DSO3062A



souhrnného přenosu  $F_A(s)$  dle vztahu (3.16) s tím rozdílem, že původně byl v proměnné  $j\omega$ . Obr. 6.1 zachycuje průběhy budicího napětí (CH1 - modrá) a napětí na svorkách syntetické impedance (CH2 - fialová). Aby bylo možné měřit tvar proudu pomocí matematické funkce v osciloskopu (Math - zelená), byl do série se syntetickou impedancí ( $\bar{Z} = 1 \text{ k}\Omega$ ) zařazen odpor s hodnotou  $1 \text{ k}\Omega$ ,  $1 \text{ V}$  tak odpovídá  $1 \text{ mA}$ . Na obr. 6.1 a) jsou tyto průběhy bez jakékoliv filtrace, tzn.  $F_A(s) = 1$ . Je vidět, že složky s vyšší frekvencí způsobily rozkmitání napětí syntetické impedance (fialová) a tím obvod v podstatě vyřadily z funkce. Po připojení dolnoproustního filtru prvního řádu se zlomovou frekvencí  $22 \text{ kHz}$  kmity vymizely a zároveň nedošlo k větší deformaci charakteristik. Průběhy obvodových veličin pak nabyly očekávaných tvarů na obr. 6.1 b).

Pro ověření základní funkce byla také syntetizována nejjednodušší možná impedance  $\bar{Z} = R$ , tj. činný odpor. Bylo provedeno několik měření při různých hodnotách  $R$  a při všech rozsazích absolutní hodnoty. Nejmenší možný odpor pro syntetizaci se ukázal  $R = 50 \Omega$ . Při nižších hodnotách výstup obvodu kmital, a tak nebyla impedance vůbec syntetizována. Toto omezení může souviset s limitací proudů výstupních zesilovačů. Případně může mít vliv impedance použitého analyzátoru, která je rovněž  $50 \Omega$ . Co se týká maximálních hodnot, ukázalo se, že při daném rozsahu  $W_i$  je

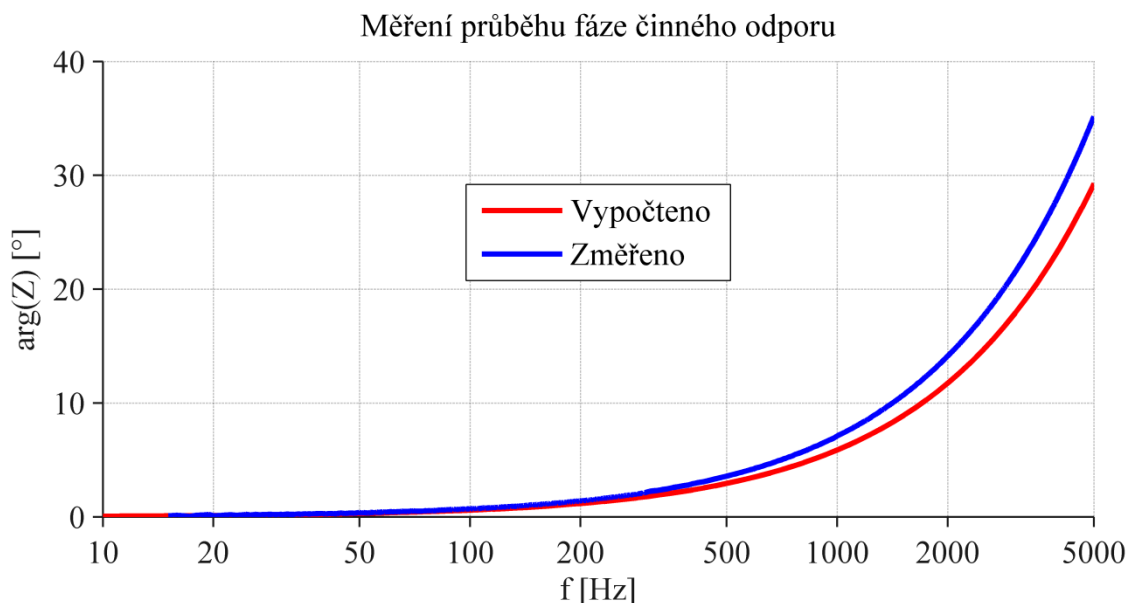


**Graf. 6.1** Průběhy absolutních hodnot syntetizovaných odporů na rozsahu  $W_2 = 100 \Omega$ , měřeno analyzátozem HP4195A



obvod schopen spolehlivě syntetizovat velikosti absolutní hodnoty v rámci dvou řádů. Např. pro rozsah  $W_2 = 100 \Omega$  se jako praktické maximum ukázala hodnota  $R = 10 k\Omega$ . Výše se již projevila značná chyba. Ukazuje to měření, které zachycuje graf. 6.1. Na zmíněném rozsahu absolutní hodnoty jsou syntetizovány hodnoty odporu od  $R = 50 \Omega$  až do  $R = 100 k\Omega$ . Při nejnižší hodnotě byl odpor  $R$  menší než  $W$ , bylo proto nutné dbát na nízkou amplitudu buzení, aby nedošlo k saturaci výstupu, protože implementovaný přenos  $F(z)$  měl vyšší absolutní hodnotu než 1. Tři nejnižší charakteristiky odpovídají zadání v celém frekvenčním rozsahu. Průběh pro  $R = 10 k\Omega$  vykazuje odchylku na frekvencích nad  $2 kHz$ . Charakteristika  $R = 100 k\Omega$  je zatížena chybou už od  $500 Hz$ .

Výsledek měření fáze činného odporu spolu s vypočteným průběhem zachycuje graf. 6.2. Implementovaný přenos při tomto pokusu měl nejjednodušší možný tvar  $F(z) = 1$ . Tvar fáze je tak ovlivněn jen vlivem zpoždění výpočtu a dodatečných filtrů, dle popisu v kapitole 3.2. Ze srovnání obou průběhů je patrná odchylka v horní části frekvenčního spektra dosahující hodnoty zhruba  $5^\circ$  pro nejvyšší frekvenci. Vzhledem k základnímu vztahu (3.7) je tato kladná odchylka fáze impedance od vypočtené způsobena větším reálným zpožděním zpracování signálu než je předpokládané. Příčinou mohou být parazitní vlastnosti použitých diskrétních součástek v přídavných filtrech i v jiných částech obvodu. Rovněž se mohou projevit zpoždění převodníků, případně může být odchylka způsobena kombinací těchto vlivů.



**Graf. 6.2 Měření průběhu fáze činného odporu, při:  $W_2 = 100 \Omega$ ,  $f_s = 100 kHz$ ,  $R = 100 \Omega$ , měřeno pomocí analyzátoru HP4195A**

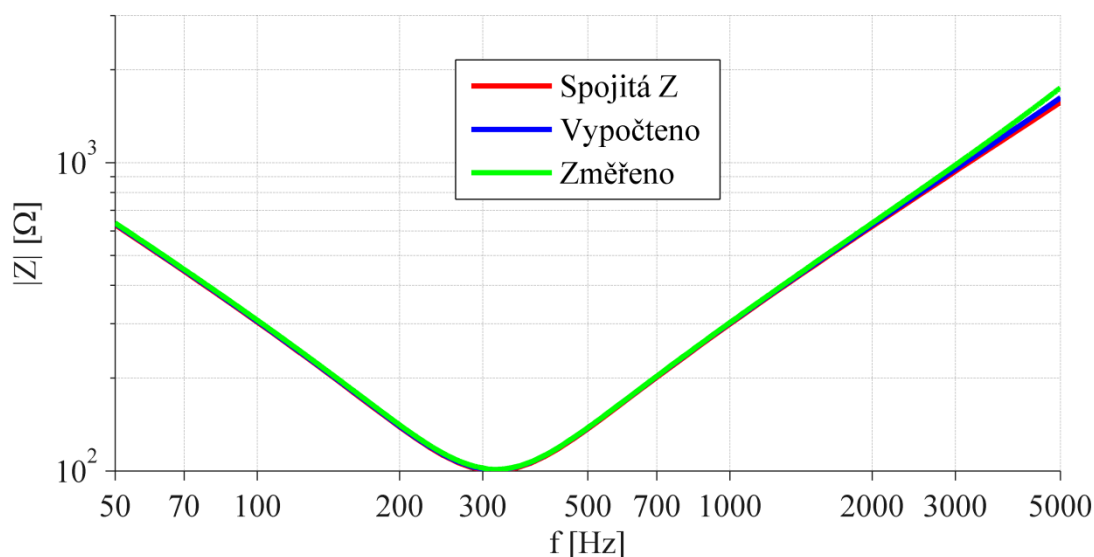


## 6.2 Měření základních impedancí

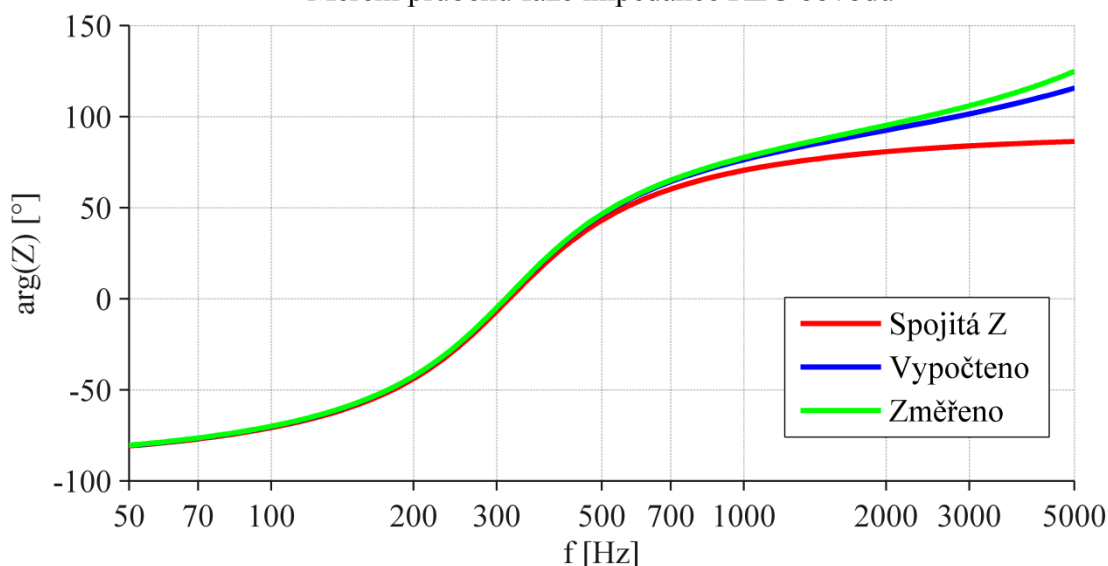
Důležitou částí experimentu bylo změření průběhů základních impedancí dle kapitoly 2.1. Nejjednodušší variantě, činnému odporu, byla věnována podstatná část předchozí kapitoly, proto se dále věnuji jen zbylým pěti typům. Graf. 6.3 ukazuje průběhy amplitudy a fáze sériového RLC obvodu, které byly syntetizovány pomocí přenosu  $F(z)$  dle vztahu (6.1). Graf. 6.3 srovnává spojitou impedanci (červená)

$$F(z) = \frac{0,0099 - 0,0099z^{-2}}{1 - 1,98z^{-1} + 0,9802z^{-2}} \quad (6.1)$$

Měření průběhu amplitudy impedance RLC obvodu



Měření průběhu fáze impedance RLC obvodu



**Graf. 6.3 Měření impedančních průběhů sériového RLC obvodu, při:  $W_2 = 100 \Omega$ ,  $f_S = 100 \text{ kHz}$ ,  $R = 100 \Omega$ ,  $L = 0,05 \text{ H}$  a  $C = 5 \mu\text{F}$ , měřeno pomocí analyzátoru HP4195A**



s teoretickou vypočtenou (modrá), která představuje diskrétní verzi se započteným vlivem doplňkových filtrů i zpoždění výpočtu. Nakonec samozřejmě i změřený průběh (zelená). Co se týče amplitudy, změřená hodnota věrně odpovídá spojitému průběhu i vypočtenému. Jen na samém konci frekvenčního rozsahu je patrná drobná odchylka. Ve fázi jsou průběhy totožné zhruba do frekvence 500 Hz. Poté se již začíná projevovat chyba fáze způsobená zpožděním výpočtu. Díky ní jsou zbylé průběhy odchýleny od spojitého. Nicméně změřené hodnoty odpovídají teoretickým vypočteným s malou odchylkou na konci frekvenčního pásma.

Vzhledem k počtu měření a rozsahu grafů jsou průběhy ostatních základních obvodů umístěny v příloze H. Dvě zapojení (sériový RC člen a kombinovaný RRC) byly změřeny rovněž se zápornou hodnotou kapacity, výsledky jsou také v příloze H. Co se týká kvality syntetizovaných impedancí, změřené průběhy absolutní hodnoty odpovídají spojitému průběhu i vypočtenému. U některých obvodů je patrná drobná odchylka na vyšších frekvencích, obdobná jakou ukazuje graf. 6.3. Lze také pozorovat drobné rušení na frekvenci 50 Hz, patrně způsobené interferencí síťového napájení. U fázových charakteristik se projevuje systematická chyba způsobená zpožděním výpočtu a také dolnoproústními filtry popsaná v kapitole 3.2. Možné příčiny této odchylky jsou diskutovány v kapitole 6.1. V podstatě je díky ní přičtena ke každému průběhu fáze funkce rostoucí s frekvencí. Vlivem zobrazení různých rozsahů úhlů na jednotlivých grafech je relativní velikost odchylky různá, často také výrazně mění tvar charakteristiky na horním konci frekvenčního pásma. Její absolutní velikost je však všude zhruba stejná. Popisovaná chyba fáze se jeví jako největší problém diskretizace základních impedancí. Má však výhodu, že ji lze matematicky modelovat, o čemž svědčí fakt, že změřené charakteristiky fáze odpovídají s malou odchylkou na vyšších frekvencích teoretickému průběhu.



### 6.3 Ověření návrhu filtru z tabulky hodnot

Naprogramované algoritmy návrhu filtru  $F(z)$  k implementaci na základě tabulky hodnot byly nejprve podrobeny jednoduchému testu. Jako zadání optimalizační úlohy při tom byly použity průběhy amplitudy a fáze jedné z impedancí popsaných v kapitole 2.1. Tímto pokusem byla testována schopnost nalézt optimální přenos  $F(z)$  v případě, kdy je jisté, že existuje. To je základní podmínka, aby algoritmy byly schopné řešit zadání, u kterých není zaručena existence odpovídajícího náhradního obvodu. K testu byla použita impedance kombinovaného RRC členu dle obr. 2.1 c). Pro srovnání vztah (6.2) ukazuje přenos  $F(z)$  vypočtený diskretizací analyticky na základě spojitě impedance. Pro srovnání představuje příklad výsledku optimalizační úlohy pro filtr IIR vztah (6.3). Rozdíl mezi oběma přenosy je minimální, v podstatě jediná drobná odchylka je v koeficientech čitatele. Prakticky jsou ale jejich průběhy totožné a měření obou vede na stejné charakteristiky, jako v příloze H pro daný typ impedance. Byla tak splněna základní podmínka funkce navrženého algoritmu, kterou je nalezení optimálního přenosu (filtru) v případě, že jeho existence je zaručena. Pro algoritmus návrhu FIR filtru byl proveden obdobný experiment s přenosem průměrovacího filtru se stejným výsledkem.

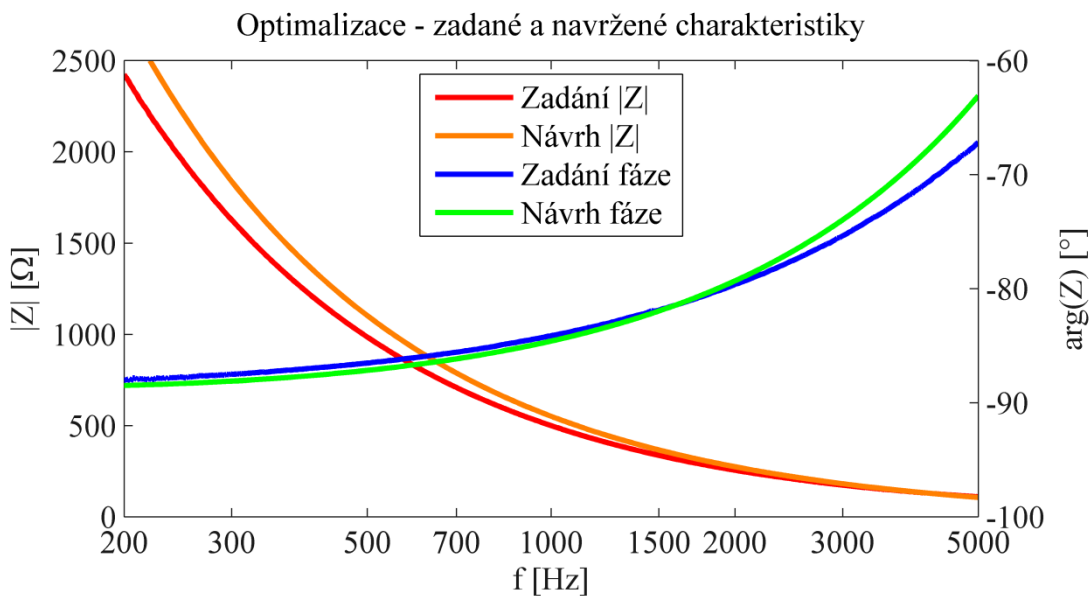
$$F(z) = \frac{0,995 - 0,9851z^{-1}}{1 - 0,9802z^{-1}} \quad (6.2)$$

$$F(z) = \frac{0,9946 - 0,9847z^{-1}}{1 - 0,9802z^{-1}} \quad (6.3)$$

Následně byl prováděn experiment s charakteristikami impedance pocházejících z experimentu Kodejšky (2013) zmíněných v kapitole 2.1. Vzhledem k praktickému omezení minimální absolutní hodnoty na  $50 \Omega$  popsanému v kapitole 6.1 musela být data upravena. V původní verzi totiž podstatná část průběhu je pod touto hodnotou. Proto byla při experimentech velikost absolutní hodnoty zvětšena do rozsahu, který je zařízení schopno bezpečně syntetizovat, průběh fáze zůstal původní. Při pokusech navrhnout filtr pro daná data se ukázala velmi důležitá volba řádu přenosu. Pro některé řády nebylo možné nalézt vhodné řešení optimalizační úlohy. Jako klíčový se ukázal také vliv počátečních hodnot pro optimalizaci. Opakované řešení úlohy při stejných počátečních datech vede na stejné, resp. velmi podobné výsledky. Proto se jako praktické ukázalo nejprve zvolit řád filtru a několikrát za sebou spustit optimalizační úlohu, pokaždé s jiným náhodně zvoleným počátečním vektorem. V případě, že některé řešení vyhovuje



přibližně zadání, je vhodné hodnoty přenosu uložit a následně je použít jako výchozí bod optimalizace, při které je možné zpřesňovat výsledek úpravou dílčích parametrů. Pokud není žádný výsledek vyhovující pro daný řád filtru, je nutné opakovat postup s jinou hodnotou. Ukázalo se, že nalézt přenos filtru  $F(z)$ , který by teoreticky dobře aproximoval průběhy dle Kodejšky (2013), je možné. Graf. 6.4 zobrazuje zadané průběhy impedance a charakteristiky, které by měly být syntetizovány pomocí navrženého přenosu. Při návrhu byla použita kompenzace fázové charakteristiky, proto není navržený průběh fáze tolik odkloněn od zadání. Do charakteristik návrhu byl započten vliv zpoždění výpočtu i filtrů v obvodu.



**Graf. 6.4** Charakteristiky navrženého filtru, upravena abs. hodnota zadání – zvětšena 20×

$$F(z) = \frac{3,32 - 0,4808z^{-1} - 3,17z^{-2} + 0,3308z^{-3}}{1 + 0,4592z^{-1} + 0,03347z^{-2} + 0,5142z^{-3}} \quad (6.4)$$

$$F(z) = \frac{0,1 + 0,2z^{-1} + 0,4z^{-2} + 0,2z^{-3} + 0,1z^{-4}}{1} \quad (6.5)$$

Navržený přenos ve tvaru (6.4) však při měření vykazoval nestabilní chování. Výstup obvodu netlumeně kmital, a tak impedance nebyla syntetizována. To vše navzdory stabilitě přenosu  $F(z)$ , který měl všechny póly bezpečně uvnitř jednotkové kružnice. Jako jedna z možných příčin se nabízela poloha nul (kořenů čitatele) tohoto přenosu. Některé z nich se nacházely na hraně stability nebo v její blízkosti. Proto bylo provedeno měření, při kterém přenos  $F(z)$  měl podobu průměrovacího filtru dle vztahu (6.5) se dvěma nulami mimo jednotkovou kružnici. Takto změřené charakteristiky však

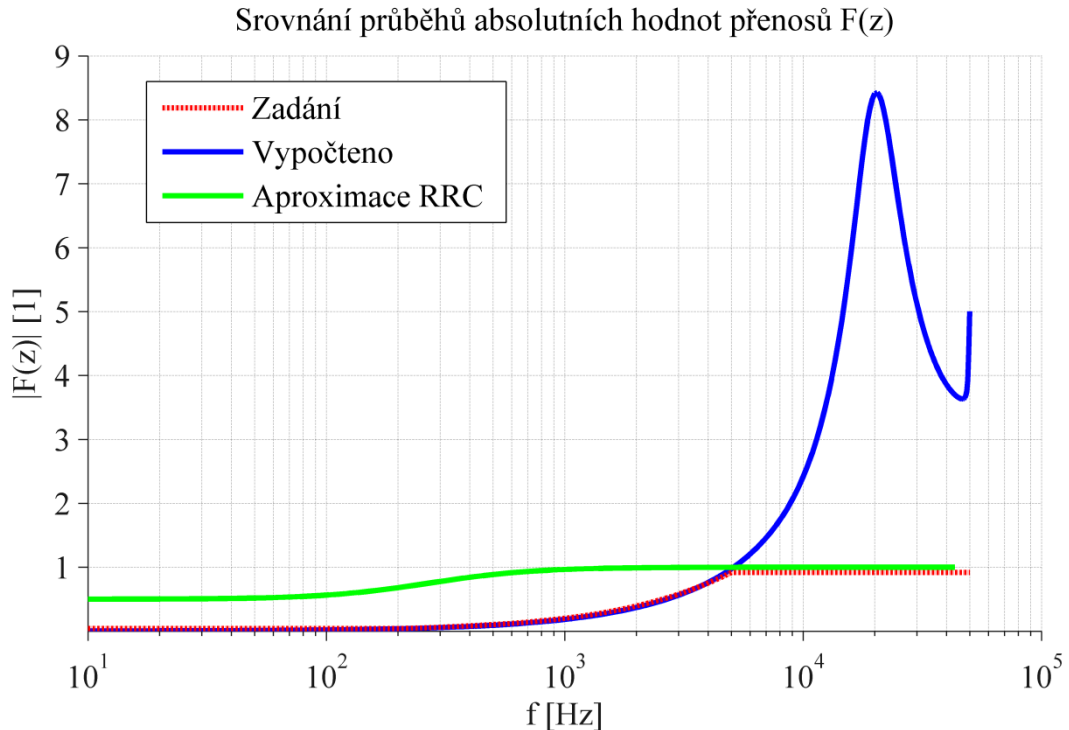




odpovídaly vypočteným a při měření se neobjevila žádná nestabilita. Obvod fungoval naprosto normálně. Možnost, že by nestabilní chování bylo způsobeno nulami vně nebo na jednotkové kružnici, se nepotvrdila.

Jako vysvětlení zjištěné nestability se nabízel také překmit v charakteristice průběhu absolutní hodnoty přenosu  $F(z)$ . Zadaný průběh absolutní hodnoty impedance ve formě vektoru  $\vec{\psi}_Z$  je přepočten na vektor zadání  $\vec{\psi}_R$  přenosu pro optimalizační úlohu dle vztahu (6.6). Podrobně je tento přepočet popsán v kapitole 3.2. Důležitý je v tomto okamžiku fakt, že klesající amplitudová charakteristika je přepočtena na rostoucí charakteristiku implementovaného přenosu podle vztahu (3.7). Graf. 6.5 zachycuje tvar zadání amplitudové charakteristiky (červená) a rovněž výsledek výpočtu (modrý). V oblasti zájmu (do 5 kHz) vypočtený průběh dobře aproximuje zadání. Na vyšších frekvencích je na něm patrný výrazný překmit přes hodnotu 1. Jak bylo výše uvedeno, pokud je absolutní hodnota přenosu  $F(z)$  vyšší než jedna, může dojít při velké amplitudě

$$\psi_{Rk} = \frac{W}{\psi_{Zk}}; \quad k = 1 \dots K \quad (6.6)$$



**Graf. 6.5** Srovnání průběhu absolutních hodnot různých přenosů  $F(z)$

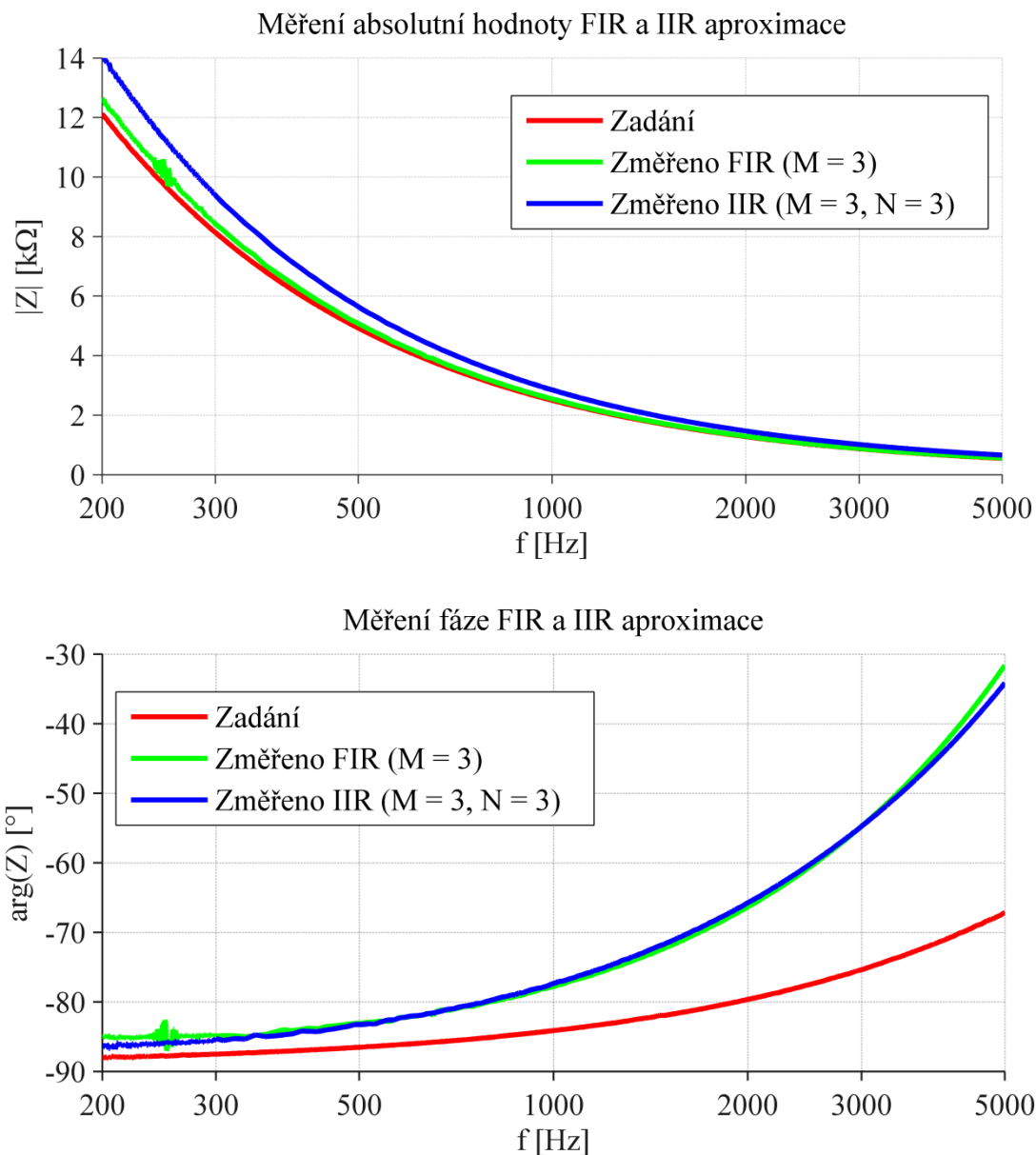


signálu k saturaci výstupu D/A převodníku a navazujících obvodů. Při průchodu signálů s vyšší frekvencí než 5 kHz dochází k výraznému zesílení jejich amplitudy. To je ale v naprostém rozporu s účelem dodatečných filtrů DP1 a DP2 popsanych v kapitole 3.2, jejichž účelem je právě odstraňovat složky signálu s vyšší frekvencí. Díky překmitu tak prakticky dochází k eliminaci těchto filtrů. Pro srovnání graf. 6.5 zachycuje i monotónní průběh absolutní hodnoty (zelený) přenosu  $F(z)$  dle vztahu (6.3), který syntetizuje průběhy RRC obvodu. Důležité je, že u tohoto průběhu se žádné nestabilní chování neobjevilo. Bylo proto provedeno několik měření, která prokázala, že přenosy bez podobného překmitu jako výše popsany vykazovaly stabilní chování dle předpokladů. Naopak přenosy s překmity kmitaly a způsobovaly nefunkčnost zařízení. Jako pravděpodobná příčina nestabilního chování se tedy jeví parazitní překmit amplitudové charakteristiky implementovaného přenosu. Za účelem odstranění výše popsaneho nežádoucího překmitu v amplitudové charakteristice bylo provedeno několik pokusů o úpravu zadání optimalizační úlohy.

Jako nejjednodušší řešení se jeví použití nižšího rozsahu absolutní hodnoty tak, aby celý průběh absolutní hodnoty implementovaného přenosu byl menší než jedna spolu s úpravou váhových koeficientů tak, aby byl překmit co nejnižší. Pro použitá data by to znamenalo nutnost rozsahu řádově menšího než  $W_1 = 10 \Omega$ , který však není k dispozici. Aby bylo možné toto řešení vyzkoušet, musel být proveden opačný postup. Amplitudová charakteristika zadané impedance proto byla vynásobena faktorem 100. Graf. 6.6 zobrazuje zadání a výsledky měření s tímto postupem. Implementační přenos byl navržen jako FIR a IIR filtr, lze tak srovnat i vlastnosti obou variant. Co se týká amplitudové charakteristiky, FIR varianta dle vztahu (6.7) vychází jednoznačně lépe, drobná odchylka od zadání je patrná jen na začátku frekvenčního pásma. Naopak IIR implementace dle vztahu (6.8) vykazuje chybu podstatně vyšší. Částečně může být odchylka způsobena tím, že byl použit rozsah absolutní hodnoty  $W_2 = 100 \Omega$  a hodnoty na začátku pásma mírně přesahují rozsah dvou řádů (důvod dle kapitoly 6.1). Nicméně tato chyba je pro obě metody stejná, FIR varianta tedy z tohoto srovnání vychází lépe.

$$F(z) = \frac{0,4469 - 0,6036z^{-1} + 0,4928z^{-2} - 0,3357z^{-3}}{1} \quad (6.7)$$

$$F(z) = \frac{0,2871 - 0,382z^{-1} + 0,002349z^{-2} + 0,09261z^{-3}}{1 - 0,9582z^{-1} + 0,1472z^{-2} - 0,01051z^{-3}} \quad (6.8)$$



**Graf. 6.6 Porovnání aproximace pomocí přenosu FIR a IIR dle optimalizace, abs. hodnota zadání zvětšena 100×, při:  $W_2 = 100 \Omega$ ,  $f_s = 100 \text{ kHz}$ , měřeno analyzátozem HP4195A**

Pokud jde o fázové průběhy obou metod, ukázalo se, že použití kompenzace fázové charakteristiky za účelem odstranění chyby zpožděním výpočtu a filtrů v obvodu není možné. Když byla kompenzace započtena do zadání optimalizační úlohy, nebylo možné nalézt vyhovující průběh. Pokud byly například koeficienty váhových matic nastaveny tak, aby zdůraznily odchylky fáze, byl sice nalezen přenos s relativně dobrou aproximací fáze, ale s naprosto nereálnými hodnotami amplitudové charakteristiky. Proto jsou obě fázové charakteristiky, které zachycuje graf. 6.6, značně vzdáleny od zadání.

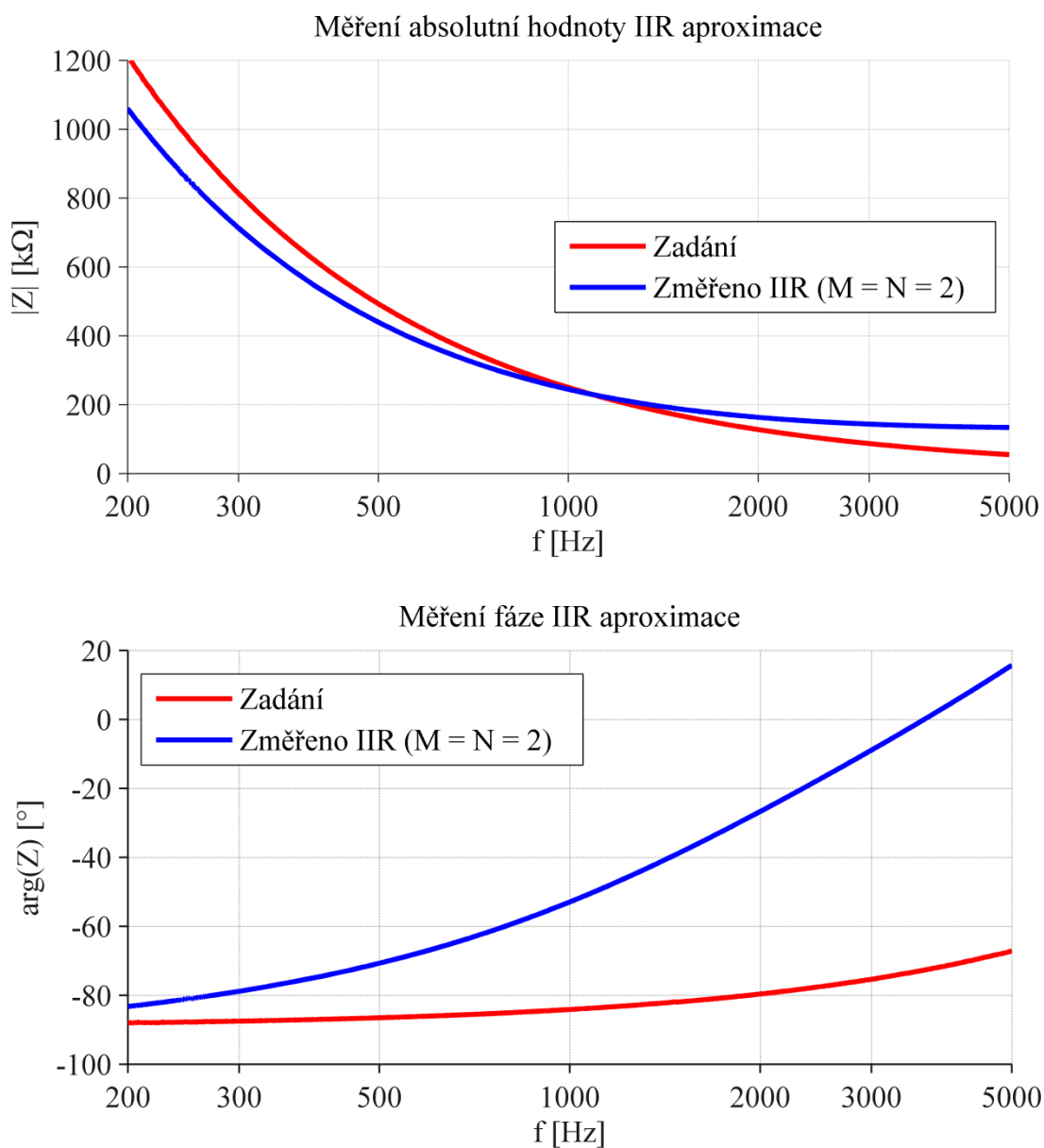


Experiment prokázal, že vhodnou volbou rozsahu absolutní hodnoty a váhových koeficientů by bylo možné nalézt funkční přenos pro amplitudu, nicméně přesnost aproximace výrazně utrpěla hlavně z hlediska průběhu fáze.

Druhou možností jak navrhnout přenos  $F(z)$ , který bude řádně syntetizovat zadané impedanční průběhy bez nestabilního kmitání, byla úprava tvaru amplitudové charakteristiky, která je brána jako zadání optimalizační úlohy. Nejprve bylo zadání amplitudy upraveno tak, že na zvolené frekvenci byl k původní charakteristice napojen polynomiální průběh, který měl na vyšších frekvencích konstantní hodnotu 1. V bodě napojení byla zajištěna rovnost nejen funkční hodnoty, ale také prvních dvou derivací. Tento postup přinesl mnoho obtíží s implementací, nejvíce problematický byl přibližný výpočet první a druhé derivace pomocí difference. Ukázalo se však, že pokud nebyl bod napojení dostatečně nízko, polynomiální průběh rovněž obsahoval výrazný překmit. Aby byl výsledek monotónní bez překmitu, musela být frekvence, ve které se napojoval polynomiální průběh, nepříjemně nízká. Metoda úpravy napojením polynomu s rovností prvních dvou derivací se ukázala jako nefunkční pro zadané průběhy. Proto byla nakonec data pro IIR verzi upravena jednoduchým způsobem.

Kolem zvoleného bodu na amplitudové charakteristice byla data středovou symetrií zobrazena z nižších frekvencí na vyšší tak, aby výsledkem byl monotónní průběh s konstantní hodnotou 1 na vyšších frekvencích. V bodě napojení tak byla zajištěna rovnost jen do první derivace. Po úpravě váhových koeficientů byl navržen přenos  $F(z)$  dle vztahu (6.9). Změřené charakteristiky impedance spolu se zadáním zachycuje graf. 6.7. Výsledky jsou obdobné jako u předchozího pokusu. Změřená amplituda je však více odchýlena od požadované. Fáze zadání nevyhovuje ze stejného důvodu jako u předchozího pokusu, jejím zdůrazněním by byl průběh amplitudy výrazně deformován.

$$F(z) = \frac{0,8347 - 0,1874z^{-1} - 0,6477z^{-2}}{1 + 0,002014z^{-1} - 0,8082z^{-2}} \quad (6.9)$$



**Graf. 6.7** Měření aproximace dat IIR filtrem, po úpravě tvaru zadání, abs. hodnota zadání zvětšena 10×, při:  $W_2 = 100 \Omega$ ,  $f_S = 100 \text{ kHz}$ , měřeno analyzátozem HP4195A



## 7 Dosažené výsledky a možnosti pokračování práce

Sestavené zařízení je dle provedených měření schopno spolu s obslužnými skripty syntetizovat impedanční průběhy s absolutní hodnotou od  $50\ \Omega$  do  $0,1\ M\Omega$ . Nepodařilo se ale dosáhnout spodní hranice zadání  $10\ \Omega$ . Při pokusech o syntézu průběhů pod  $50\ \Omega$  výstup obvodu kmital, což vedlo k nefunkčnosti zařízení. Jako vysvětlení se nabízí velikost výstupní impedance použitého analyzátoru (HP4195A), která je právě  $50\ \Omega$ . Stejnou hodnotu má i generátor (HP33120A) použitý při měření funkčních průběhů na osciloskopu. V další práci by bylo vhodné prozkoumat vliv poměru absolutní hodnoty syntetizované impedance a impedance připojeného zdroje signálu na funkci zařízení v souvislosti s omezením výstupního napětí a proudu použitých operačních zesilovačů.

Z provedených experimentů vyplývá, že lze syntetizovat spolehlivě průběhy absolutní hodnoty impedance. Ve fázových průbězích se však vyskytuje systematická chyba, která je způsobena zpožděním reálného výpočtu a také doplňkových filtrů v obvodu. Tuto chybu nelze úplně odstranit, jen omezit. V současné době je v Laboratoři tlumení vibrací potřeba řešit syntézu impedance tak, že charakteristiky amplitudy a fáze tlumicího piezoelementu a obvodu syntetické impedance se protínají na jedné frekvenci. Vzhledem k tomu, že chybu fáze lze matematicky modelovat, může být zahrnuta do výpočtu takové úlohy. Vizí je shoda průběhů v širším frekvenčním rozsahu, proto by v pokračování práce bylo vhodné upravit obvod pro vyšší vzorkovací frekvenci. Tím by se redukovalo zpoždění výpočtu a zároveň by bylo možné použít doplňkové filtry s vyšší zlomovou frekvencí, což by vedlo k menším odchylkám zadané a syntetizované fáze.

Návrh implementačních filtrů na základě tabulky hodnot absolutní hodnoty a fáze impedance se jeví jako velmi široké téma. Ukázalo se, že nalézt přenos pro optimální syntetizaci průběhu absolutní hodnoty je možné. Dosažení dobré shody ve fázi se však jeví jako podstatně komplikovanější. Byl nalezen teoreticky optimální přenos, který se ovšem při měření ukázal jako nefunkční. Úpravou zadání je možné navrhnout funkční přenos, nicméně za cenu fatální chyby fázové charakteristiky. Proto by bylo vhodné v navazující práci analyzovat podrobněji řešení optimalizační úlohy hlavně z hlediska existence přenosů pro zadané tvary charakteristik.



## 8 Závěr

Prostudoval jsem literaturu zabývající se realizací syntetických impedancí, typy a parametry impedancí, které je potřeba syntetizovat. Následně jsem provedl rešerši použitelných metod pro digitální realizaci a rovněž přehled několika přístupů pro diskretizaci spojitých impedancí. Na základě výsledků rešerše byly stanoveny teoretické požadavky na implementační metodu a vybrána vhodná varianta k praktické realizaci. Zvolena byla topologie *obvodu syntetické admitance* s tím, že jádrem digitální části je mikropočítač typu ARM. Ze sady diskretizačních metod byla na základě analýzy zvolena Tustinova aproximace. Pro návrh přenosu k syntetizaci impedančních průběhů zadaných tabulkou byla vybrána metoda numerické optimalizace minimalizací vhodného kritéria.

Hardware použitý pro výsledné zařízení je složen z několika modulů na jednotlivých deskách plošných spojů. Pro digitální modul a část desky převodníků bylo použito hotové řešení s drobnými modifikacemi. Zbytek, především analogový výstupní modul, musel být vytvořen, protože nebylo dostupné vhodné existující řešení. Zadaný rozsah absolutní hodnoty byl rozdělen do čtyř dekad. Zařízení komunikuje s nadřazeným řídicím systémem pomocí virtuálního sériového portu. Fyzicky je propojení realizováno USB portem. Vyvinul jsem firmware mikropočítače umožňující syntézu impedance dle teoretických vztahů. Rovněž jsem vyvinul obslužné programy v počítači, které zajišťují komunikaci se zařízením a implementují návrh filtrů diskretizací spojitě impedance nebo návrh numerickou optimalizací. Postup obou typů návrhu je dokumentován.

Provedl jsem na zvolených příkladech impedancí experimentální srovnání teoretických, vypočtených a změřených průběhů. Z měření průběhů diskrétních verzí základních impedancí vyplývá, že zařízení je schopno spolehlivě syntetizovat amplitudové průběhy od hodnoty  $50\ \Omega$  do  $0,1\ M\Omega$ . Fázové charakteristiky jsou zatíženy systematickou odchylkou, která je dána zpožděním výpočtu a vlastnostmi přídavných filtrů v obvodu a projevuje se výrazněji na vyšších frekvencích. Výhodou této chyby je, že je deterministická a lze ji matematicky modelovat. U algoritmů pro syntetizaci impedance na základě tabulky hodnot byla prokázána jejich schopnost nalézt optimální přenos. Protože se objevily komplikace s jejich reálnou implementací, byly použity korekce zadání, které zajistily funkčnost přenosů, ale zhoršily vlastnosti průběhů, především kriticky ovlivnily tvar fáze. Možnosti pokračování práce jsou diskutovány ve zvláštní kapitole.



## Použitá literatura

- AD5781, 2010. [online]. Analog Devices [cit. 2013-11-25], rev. D. Dostupné z: [http://www.analog.com/static/imported-files/data\\_sheets/AD5781.pdf](http://www.analog.com/static/imported-files/data_sheets/AD5781.pdf).
- AD7634, 2007. [online]. Analog Devices [cit. 2013-11-25], rev. B. Dostupné z: [http://www.analog.com/static/imported-files/data\\_sheets/AD7634.pdf](http://www.analog.com/static/imported-files/data_sheets/AD7634.pdf).
- ARM. 2010. *Cortex-M4 Technical Reference Manual* [online]. ARM Limited [cit. 2013-11-20], rev. r0p0. Dostupné z: [http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439b/DDI0439B\\_cortex\\_m4\\_r0p0\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439b/DDI0439B_cortex_m4_r0p0_trm.pdf).
- BEHRENS, Sam, Andrew FLEMING a Reza MOHEIMANI. 2003. *Vibrational damping apparatus and method for deriving a digital signal processing algorithm* [patent online]. Vynález, US20030057807 A1. Uděleno 27. března 2003 [cit. 2013-11-13]. Dostupné z: <http://www.google.com/patents/US20030057807>.
- FLEMING, Andrew J. 2004. *Synthesis and Implementation of Sensor-less Shunt Controllers for Piezoelectric and Electromagnetic Vibration Control*. [Cit. 2013-11-13]. Dizertační práce. The University of Newcastle, School of Electrical Engineering and Computer Science. Vedoucí práce S. O. Reza Moheimani. Dostupné z: <http://hdl.handle.net/1959.13/32006>.
- FLEMING, Andrew J., Sam BEHRENS a S. O. Reza MOHEIMANI. 2000. Synthetic impedance for implementation of piezoelectric shunt-damping circuits. In: *Electronics Letters* [online]. Stevenage: Institution of Engineering and Technology, [cit. 2013-11-10], vol. 36, no. 18, s. 1525, 1526. ISSN: 0013-5194. DOI: [10.1049/el:20001083](https://doi.org/10.1049/el:20001083).
- FLEMING, Andrew J., Sam BEHRENS a S. O. Reza MOHEIMANI. 2002. Optimization and Implementation of Multimode Piezoelectric Shunt Damping Systems. In: *IEEE/ASME Transactions on Mechatronics* [online]. New York: IEEE Robotics and Automation Society, [cit. 2013-11-10], vol. 7, no. 1, s. 87-94. ISSN 1083-4435. DOI: [10.1109/3516.990891](https://doi.org/10.1109/3516.990891).
- FORSYTHE W. [William]. a R. M. [Roger Morgan] GOODALL. 1991. *Digital Control: Fundamentals Theory and Practise* [online]. London: MACMILLAN EDUCATION LTD., [cit. 2013-11-23]. ISBN 0-07-021600-2. At: Ústav mechatroniky a technické informatiky TUL.
- GIANNI, Carmine aj. 2007. The Universal Circuit Simulator: A Mixed-Signal Approach to n-Port Network and Impedance Synthesis. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* [online]. New York: IEEE Circuits and Systems Society, [cit. 2013-11-07], vol. 54, no. 10, s. 2178-2183. ISSN 1549-8328. DOI: [10.1109/TCSI.2007.904594](https://doi.org/10.1109/TCSI.2007.904594).





KODEJŠKA, Miloš. 2013. *Návrh systémů pro potlačení hluku a vibrací s využitím piezoelektrických materiálů*. [Cit 2013-11-01, v průběhu práce před vydáním]. Dizertační práce. Technická univerzita v Liberci, Fakulta mechatroniky informatiky a mezioborových studií. Vedoucí práce: Pavel Mokrý. At: Ústav mechatroniky a technické informatiky TUL.

NEČÁSEK, Jakub. 2014. *Impedanční analyzátor pro laboratoř aktivního tlumení vibrací*. [Cit 2013-11-20, v průběhu práce před vydáním] Diplomová práce. Technická univerzita v Liberci, Fakulta mechatroniky informatiky a mezioborových studií. Vedoucí práce: Pavel Márton. At: Ústav mechatroniky a technické informatiky TUL.

SLUKA, Tomáš. 2007. *Noise and vibration control using piezoelectric elements shunted by a negative capacitor*. [Cit. 2013-10-20]. Dizertační práce. Technická univerzita v Liberci, Fakulta mechatroniky informatiky a mezioborových studií. Vedoucí práce Pavel Mokrý. At: Knihovna Technické univerzity v Liberci.



## **Příloha A Obsah přiloženého CD**

**DP\_JiriBudasz.pdf** – Text práce ve formátu pdf.

**Atollic\_firmwareARM** – Složka s projektem v programu Atollic True Studio obsahující kódy firmware.

**EAGLE\_vykresy\_DPS** – Složka s projekty v programu EAGLE obsahující schémata zapojení a výkresy použitých desek plošných spojů.

**MATLAB\_skripty** – Složka obsahující použité skripty MATLABu v souborech typu m-file.



## Příloha B Odvození diskretizační metody mapování pólů

Jednou ze zvažovaných diskretizačních metod byla metoda mapování pólů popsaná Forsythem (1991). Jak bylo v textu uvedeno, použil jsem oproti původnímu zdroji jiné značení a bylo nutné provést obecnější odvození. Upravený postup dle Forsytha (1991) v mnou zvolené symbolice zde uvádím. Výchozím bodem je spojitý přenos  $F(s)$  dle vztahu (B.1). Cílem je pak diskrétní přenos  $F(z)$  dle (B.2). Proměnná  $U(s)$  představuje spojitý obraz vstupu filtru popsaného  $F(s)$  a  $Y(s)$  pak výstup. V diskrétní oblasti jsou obrazy  $U(z)$  a  $Y(z)$ . Tyto veličiny nejsou v textu práce zmíněny, protože pro praktické užití metody nejsou k výpočtu potřeba, slouží jen pro účely odvození.

$$F(s) = \frac{h_L s^L + h_{L-1} s^{L-1} + \dots + h_1 s + h_0}{d_N s^N + d_{N-1} s^{N-1} + \dots + d_1 s + d_0} = \frac{Y(s)}{U(s)} \quad (\text{B.1})$$

$$F(z) = \frac{b_0 + b_1 z^{-1} \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{1 + a_1 z^{-1} \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} = \frac{Y(z)}{U(z)} \quad (\text{B.2})$$

Nejprve jsou vypočteny póly přenosu  $s_i$ . Dle vztahu (B.3) jsou následně přepočteny do  $z$ -oblasti na diskrétní póly  $\alpha_i$ . Tento postup je nazýván mapování, odtud název metody. Z vypočtených hodnot je následně dosazením sestaven polynom  $p_D$  jmenovatele diskrétního přenosu ve tvaru (B.4), který je následně přepočtený do formy (B.5). Tím je jednoznačně určen diskrétní jmenovatel, který má stejný počet pólů jako spojitý, a tak i stejný řád  $N$ . Řád čitatele  $M$  lze volit stejný jako  $N$  nebo vyšší, nejlepší volbou je  $M = N$ , protože v případě  $M > N$  je do výpočtu zaneseno další zpoždění.

$$\alpha_i = e^{s_i}; i = 1 \dots N \quad (\text{B.3})$$

$$p_D(z) = (1 - \alpha_1 z^{-1})(1 - \alpha_2 z^{-1}) \dots (1 - \alpha_N z^{-1}) \quad (\text{B.4})$$

$$p_D(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N} \quad (\text{B.5})$$

Výpočet čitatele již není tak přímočarý. K odvození vztahu pro jeho výpočet je nejprve potřeba převést vztah (B.2) do časové oblasti na rovnici (B.6) při uvažování nulových počátečních podmínek. Diferenční rovnice (B.6) vyjadřuje hodnoty vstupu  $u(t)$  a výstupu  $y(t)$  v diskrétních okamžicích s krokem  $T$ .

$$b_0 u(t) + b_1 u(t - T) + b_2 u(t - 2T) + \dots + b_M u(t - MT) = y(t) + a_1 y(t - T) + a_2 y(t - 2T) + \dots + a_N y(t - NT) \quad (\text{B.6})$$



Vzhledem k tomu, že následující vztahy by byly v čistě obecné formě velmi komplikované a nepřehledné, rozhodl jsem se provést odvození pro hodnoty  $L = N = 3$ . Což je vzhledem k uvažovaným typům impedancí dostatečné a zároveň přehledné pro popis. Členy rovnice (B.6) jsou rozvinuty pomocí Taylorova rozvoje. Pro větší přehlednost je ze zápisu vynechána závislost veličin na čase, jsou zapsány ve zjednodušené formě,  $u(t) \equiv u$  a  $y(t) \equiv y$ . Na levé straně rovnice jsou členy rozvinuty do řádu  $M$  a na pravé potom do řádu  $N$ . Výsledek po přeskupení tak, aby u každé derivace příslušné proměnné byl vlastní koeficient, zachycuje vztah (B.7).

$$\begin{aligned}
& u(b_0 + b_1 + b_2 + b_3) + \\
& + u' \left( -b_1 \frac{T}{1!} - b_2 \frac{2T}{1!} - b_3 \frac{3T}{1!} \right) + \\
& + u'' \left( b_1 \frac{T^2}{2!} + b_2 \frac{(2T)^2}{2!} + b_3 \frac{(3T)^2}{2!} \right) + \\
& + u^{(3)} \left( -b_1 \frac{T^3}{3!} - b_2 \frac{(2T)^3}{3!} - b_3 \frac{(3T)^3}{3!} \right) = \\
& = y(1 + a_1 + a_2 + a_3) + \\
& + y' \left( -a_1 \frac{T}{1!} - a_2 \frac{2T}{1!} - a_3 \frac{3T}{1!} \right) + \\
& + y'' \left( a_1 \frac{T^2}{2!} + a_2 \frac{(2T)^2}{2!} + a_3 \frac{(3T)^2}{2!} \right) + \\
& + y^{(3)} \left( -a_1 \frac{T^3}{3!} - a_2 \frac{(2T)^3}{3!} - a_3 \frac{(3T)^3}{3!} \right)
\end{aligned} \tag{B.7}$$

Přenos (B.1) musí být pro odvození doplněn do tvaru (B.8) neznámým pomocným přenosem. Řád jeho polynomu, který tvoří čitatele i jmenovatele, je  $M$ . Vztah (B.8) je převeden do časové oblasti. Forsythe (1991) tento krok nepopisuje podrobně, ale ze souvislostí vyplývá, že před převodem jsou polynomy přenosu  $F_P(s)$  mezi sebou vynásobeny. Členy s vyšší mocninou než  $M$  jsou zanedbány, v diferenciální rovnici se tak neobjeví vyšší derivace než právě  $M$ . Výslednou rovnici zachycuje vztah (B.9).

$$F_P(s) = \frac{h_3 s^3 + h_2 s^2 + h_1 s + h_0}{d_3 s^3 + d_2 s^2 + d_1 s + d_0} \cdot \frac{f_3 s^3 + f_2 s^2 + f_1 s + f_0}{f_3 s^3 + f_2 s^2 + f_1 s + f_0} = \frac{Y(s)}{U(s)} \tag{B.8}$$



$$\begin{aligned}
 &uf_0h_0 + u'(f_0h_1 + f_1h_0) + u''(f_0h_2 + f_1h_1 + f_2h_0) + \\
 &\quad + u^{(3)}(f_0h_3 + f_1h_2 + f_2h_1 + f_3h_0) = \\
 &= yf_0d_0 + y'(f_0d_1 + f_1d_0) + y''(f_0d_2 + f_1d_1 + f_2d_0) + \\
 &\quad + y^{(3)}(f_0d_3 + f_1d_2 + f_2d_1 + f_3d_0)
 \end{aligned} \tag{B.9}$$

V tomto okamžiku jsou k dispozici dvě diferenciální rovnice, (B.7) a (B.9). Jedna vycházející z diskrétního popisu a druhá ze spojitého. Obě obsahují stejné derivace funkcí  $u$  a  $y$ . Proto jsou porovnány koeficienty jednotlivých derivací. Porovnáním jednotlivých koeficientů pravých stran rovnic vzniká lineární soustava rovnic (B.10). Obdobně pro levé strany vztah (B.11). Obě soustavy jsou pak uspořádány do maticových tvarů (B.12) a (B.13). Celkem osm rovnic obsahuje osm neznámých ( $f_0, \dots, f_3, b_0, \dots, b_3$ ). Koeficienty  $a_1, a_2, a_3$  jsou známé z první fáze výpočtu, z mapování pólů.

$$\begin{aligned}
 &1 + a_1 + a_2 + a_3 = f_0d_0 \\
 &-a_1 \frac{T}{1!} - a_2 \frac{2T}{1!} - a_3 \frac{3T}{1!} = f_0d_1 + f_1d_0 \\
 &a_1 \frac{T^2}{2!} + a_2 \frac{(2T)^2}{2!} + a_3 \frac{(3T)^2}{2!} = f_0d_2 + f_1d_1 + f_2d_0 \\
 &-a_1 \frac{T^3}{3!} - a_2 \frac{(2T)^3}{3!} - a_3 \frac{(3T)^3}{3!} = f_0d_3 + f_1d_2 + f_2d_1 + f_3d_0
 \end{aligned} \tag{B.10}$$

$$\begin{aligned}
 &b_0 + b_1 + b_2 + b_3 = f_0h_0 \\
 &-b_1 \frac{T}{1!} - b_2 \frac{2T}{1!} - b_3 \frac{3T}{1!} = f_0h_1 + f_1h_0 \\
 &b_1 \frac{T^2}{2!} + b_2 \frac{(2T)^2}{2!} + b_3 \frac{(3T)^2}{2!} = f_0h_2 + f_1h_1 + f_2h_0 \\
 &-b_1 \frac{T^3}{3!} - b_2 \frac{(2T)^3}{3!} - b_3 \frac{(3T)^3}{3!} = f_0h_3 + f_1h_2 + f_2h_1 + f_3h_0
 \end{aligned} \tag{B.11}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -\frac{T}{1!} & -\frac{2T}{1!} & -\frac{3T}{1!} \\ 0 & \frac{T^2}{2!} & \frac{(2T)^2}{2!} & \frac{(3T)^2}{2!} \\ 0 & -\frac{T^3}{3!} & -\frac{(2T)^3}{3!} & -\frac{(3T)^3}{3!} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} d_0 & 0 & 0 & 0 \\ d_1 & d_0 & 0 & 0 \\ d_2 & d_1 & d_0 & 0 \\ d_3 & d_2 & d_1 & d_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{B.12}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -\frac{T}{1!} & -\frac{2T}{1!} & -\frac{3T}{1!} \\ 0 & \frac{T^2}{2!} & \frac{(2T)^2}{2!} & \frac{(3T)^2}{2!} \\ 0 & -\frac{T^3}{3!} & -\frac{(2T)^3}{3!} & -\frac{(3T)^3}{3!} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ h_3 & h_2 & h_1 & h_0 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{B.13}$$



Obecně lze obě soustavy vyjádřit vztahy (B.14) a (B.15). Nyní je potřeba eliminovat vektor  $\vec{f}$  koeficientů pomocného polynomu a vyjádřit vektor čitatele diskrétního přenosu  $\vec{b}$ . Z rovnice (B.14) je vyjádřen vektor  $\vec{f}$  a dosazen do (B.15). Po vynásobení maticí  $\mathbf{V}^{-1}$  je výsledek ve tvaru (B.16), což je vztah pro praktický výpočet čitatele.

$$\mathbf{C}\vec{a} = \mathbf{D}\vec{f} \quad (\text{B.14})$$

$$\mathbf{V}\vec{b} = \mathbf{H}\vec{f} \quad (\text{B.15})$$

$$\vec{b} = \mathbf{V}^{-1}\mathbf{H}\mathbf{D}^{-1}\mathbf{C}\vec{a} \quad (\text{B.16})$$

K implementaci výpočtu je potřeba tvořit matice automaticky. Obecný tvar čtvercové matice  $\mathbf{D}$  zobrazuje (B.17), její rozměr je  $M + 1$ . Je tvořena koeficienty polynomu jmenovatele spojitého přenosu. Jedná se o dolní trojúhelníkovou matici, díky tomu je v případě nenulovosti  $d_0$  regulární a existuje její inverze potřebná pro výpočet. V obecném případě  $M > N$  je levý dolní roh vyplněn nulami. V reálné variantě  $M = N$  je v tomto rohu  $d_N$ . Matice  $\mathbf{H}$  je obdobně tvořena koeficienty čitatele spojitého přenosu v obecném tvaru (B.18). Rozměr je také  $M + 1$  a rovněž je čtvercová.

$$\mathbf{D} = \begin{bmatrix} d_0 & 0 & 0 & \dots & 0 & 0 \\ d_1 & d_0 & 0 & \dots & 0 & 0 \\ \dots & d_1 & d_0 & \dots & 0 & 0 \\ d_N & \dots & d_1 & \dots & \dots & \dots \\ \dots & d_N & \dots & \dots & d_0 & 0 \\ 0 & \dots & d_N & \dots & d_1 & d_0 \end{bmatrix} \quad (\text{B.17})$$

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & 0 & \dots & 0 & 0 \\ h_1 & h_0 & 0 & \dots & 0 & 0 \\ \dots & h_1 & h_0 & \dots & 0 & 0 \\ h_L & \dots & h_1 & \dots & \dots & \dots \\ \dots & h_L & \dots & \dots & h_0 & 0 \\ 0 & \dots & h_L & \dots & h_1 & h_0 \end{bmatrix} \quad (\text{B.18})$$

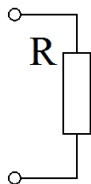
Konstantní matice  $\mathbf{C}$  a  $\mathbf{V}$  mají stejné prvky. V obecném případě se liší jen rozměry. Matice  $\mathbf{C}$  má  $N + 1$  sloupců a  $M + 1$  řádků.  $\mathbf{V}$  je čtvercová s rozměrem  $M + 1$ . Prakticky je reálná volba  $M = N$ , pak jsou matice  $\mathbf{C}$  a  $\mathbf{V}$  totožné. Jednotlivé prvky matic lze vypočítat dle vztahu (B.19). Index  $i$  představuje pořadí řádku a  $j$  sloupce.

$$\mathbf{V}_{ij} = \mathbf{C}_{ij} = \frac{[-T \cdot (j - 1)]^{(i-1)}}{(i - 1)!} \quad (\text{B.19})$$



## Příloha C Diskretizace základních impedancí

### Činný R

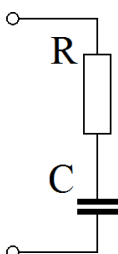


$$Z(s) = R$$

$$Z(z) = R$$

$$F(z) = \frac{W}{R}$$

### Sériový RC obvod



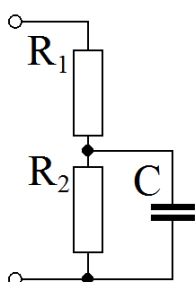
$$Z(s) = \frac{sRC + 1}{sC}$$

$$Z(z) = \frac{1 - \frac{T - 2RC}{T + 2RC} z^{-1}}{\frac{2C}{T + 2RC} (1 - z^{-1})}$$

$$F(z) = \frac{W \frac{2C}{T + 2RC} (1 - z^{-1})}{1 - \frac{T - 2RC}{T + 2RC} z^{-1}}$$

$$n_1 = -\frac{1}{RC}; RC > 0$$

### Kombinovaný RRC obvod



$$Z(s) = \frac{sR_1R_2C + R_1 + R_2}{sR_2C + 1}$$

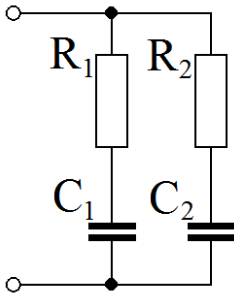
$$Z(z) = \frac{1 + \frac{TR_1 + TR_2 - 2R_1R_2C}{2R_1R_2C + TR_1 + TR_2} z^{-1}}{\frac{T + 2R_2C}{2R_1R_2C + TR_1 + TR_2} + \frac{T - 2R_2C}{2R_1R_2C + TR_1 + TR_2} z^{-1}}$$

$$F(z) = \frac{W \left( \frac{T + 2R_2C}{2R_1R_2C + TR_1 + TR_2} + \frac{T - 2R_2C}{2R_1R_2C + TR_1 + TR_2} z^{-1} \right)}{1 + \frac{TR_1 + TR_2 - 2R_1R_2C}{2R_1R_2C + TR_1 + TR_2} z^{-1}}$$

$$n_1 = -\frac{R_1 + R_2}{R_1R_2C}; \frac{R_1 + R_2}{R_1R_2C} > 0$$



### Kombinace dvou RC obvodů



$$Z(s) = \frac{s^2 R_1 R_2 C_1 C_2 + s(R_1 C_1 + R_2 C_2 + 1)}{s^2 C_1 C_2 (R_1 + R_2) + s(C_1 + C_2)}$$

$$Z(z) = \frac{W(1 + a_1 z^{-1} + b_2 z^{-2})}{b_0 + b_1 z^{-1} + b_2 z^{-2}}$$

$$F(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + b_2 z^{-2}}$$

$$a_1 = \frac{2T^2 - 8R_1 R_2 C_1 C_2}{4R_1 R_2 C_1 C_2 + 2T(R_1 C_1 + R_2 C_2) + T^2}$$

$$a_2 = \frac{4R_1 R_2 C_1 C_2 - 2T(R_1 C_1 + R_2 C_2) + T^2}{4R_1 R_2 C_1 C_2 + 2T(R_1 C_1 + R_2 C_2) + T^2}$$

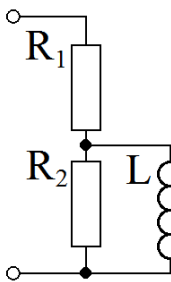
$$b_0 = W \frac{4C_1 C_2 (R_1 + R_2) + 2T(C_1 + C_2)}{4R_1 R_2 C_1 C_2 + 2T(R_1 C_1 + R_2 C_2) + T^2}$$

$$b_1 = W \frac{-8C_1 C_2 (R_1 + R_2)}{4R_1 R_2 C_1 C_2 + 2T(R_1 C_1 + R_2 C_2) + T^2}$$

$$b_2 = W \frac{4C_1 C_2 (R_1 + R_2) - 2T(C_1 + C_2)}{4R_1 R_2 C_1 C_2 + 2T(R_1 C_1 + R_2 C_2) + T^2}$$

$$n_1 = -\frac{1}{R_1 C_1}; n_2 = -\frac{1}{R_2 C_2}; R_1 C_1 > 0 \wedge R_2 C_2 > 0$$

### Kombinovaný RRL obvod



$$Z(s) = \frac{sL(R_1 + R_2) + R_1 R_2}{sL + R_2}$$

$$Z(z) = \frac{1 + \frac{R_1 R_2 T - 2L(R_1 + R_2)}{R_1 R_2 T + 2L(R_1 + R_2)} z^{-1}}{\frac{2L + R_2 T}{R_1 R_2 T + 2L(R_1 + R_2)} + \frac{2L - R_2 T}{R_1 R_2 T + 2L(R_1 + R_2)} z^{-1}}$$

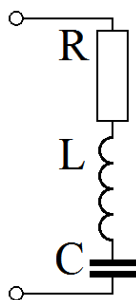
$$F(z) = \frac{W \left( \frac{2L + R_2 T}{R_1 R_2 T + 2L(R_1 + R_2)} + \frac{2L - R_2 T}{R_1 R_2 T + 2L(R_1 + R_2)} z^{-1} \right)}{1 + \frac{R_1 R_2 T - 2L(R_1 + R_2)}{R_1 R_2 T + 2L(R_1 + R_2)} z^{-1}}$$

$$n_1 = -\frac{R_1 R_2}{L(R_1 + R_2)}; \frac{R_1 R_2}{L(R_1 + R_2)} > 0$$





### Sériový RLC obvod



$$Z(s) = \frac{s^2 LC + sRC + 1}{sC}$$

$$Z(z) = \frac{W(1 + a_1 z^{-1} + b_2 z^{-2})}{b_0 + b_1 z^{-1} + b_2 z^{-2}}$$

$$F(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + b_2 z^{-2}}$$

$$a_1 = \frac{2T^2 - 8LC}{4LC + 2CRT + T^2}$$

$$a_2 = \frac{4LC - 2CRT + T^2}{4LC + 2CRT + T^2}$$

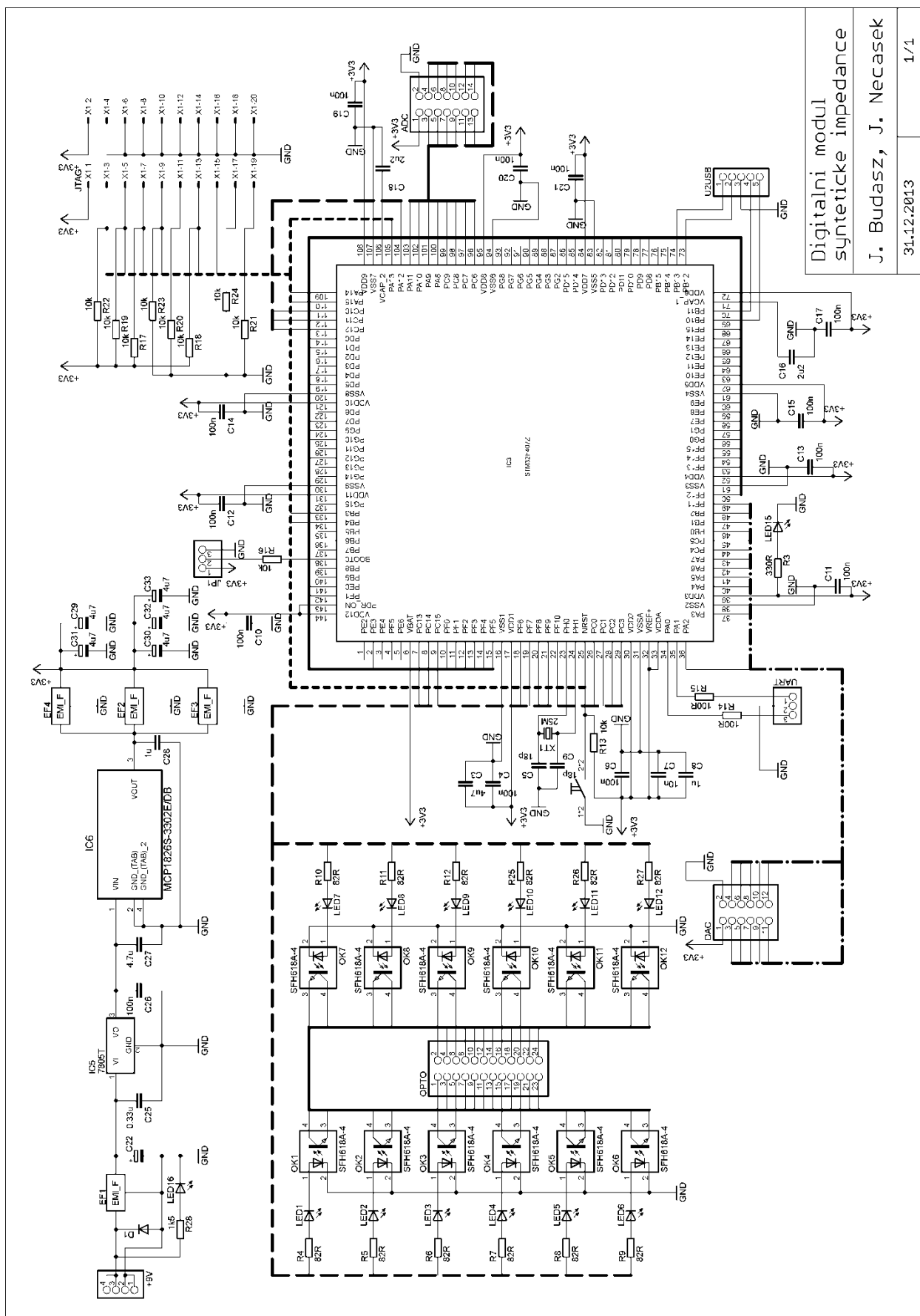
$$b_0 = W \frac{2CT}{4LC + 2CRT + T^2}$$

$$b_1 = 0$$

$$b_2 = b_0 = W \frac{2CT}{4LC + 2CRT + T^2}$$

$$Re \left\{ \frac{-RC \pm j\sqrt{4LC - R^2 C^2}}{2LC} \right\} < 0$$

## Příloha D Schéma digitálního modulu

Digitální modul  
syntetické impedance

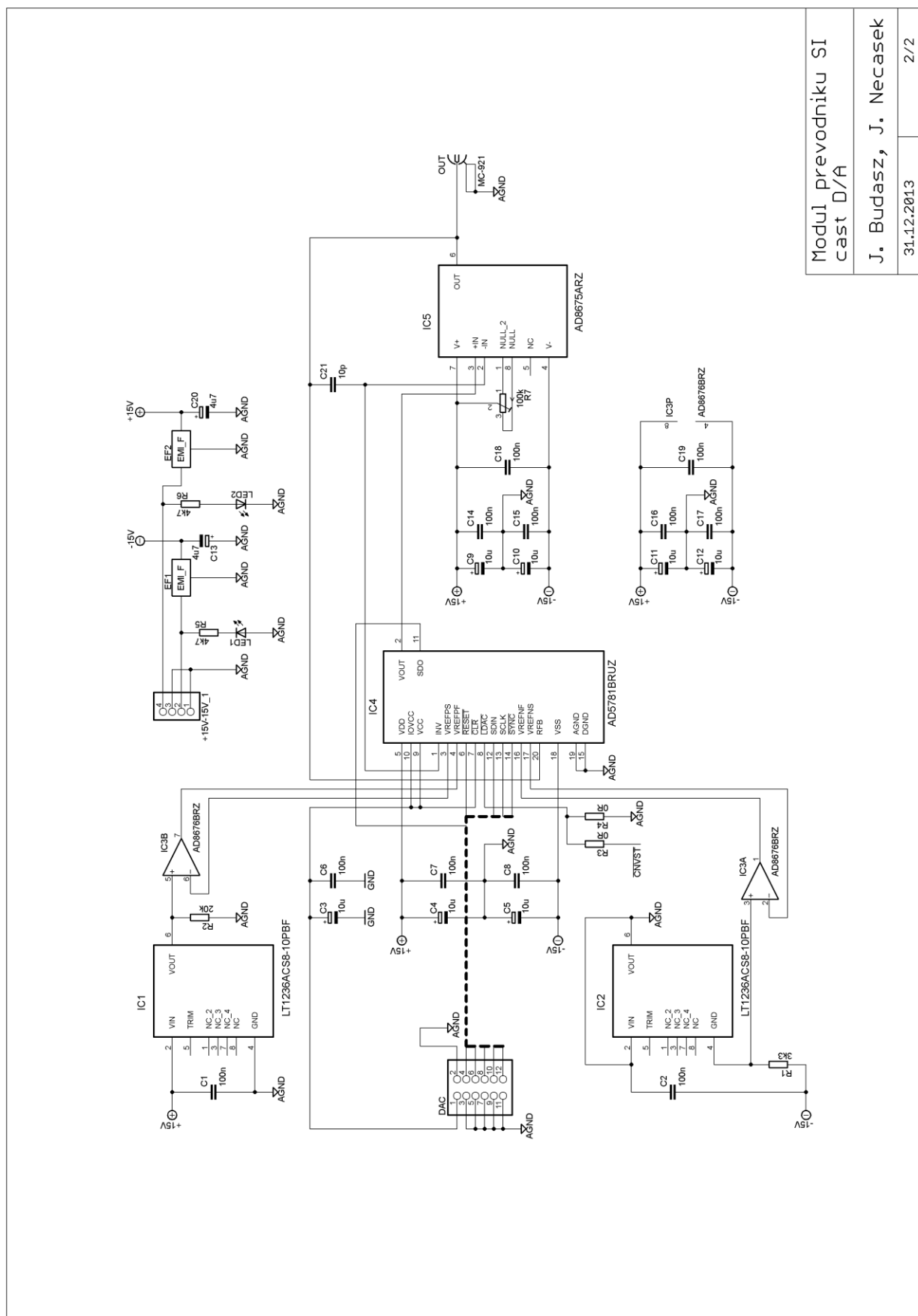
J. Budasz, J. Necasek

31.12.2013

1/1

## Příloha E Schéma modulu převodníků





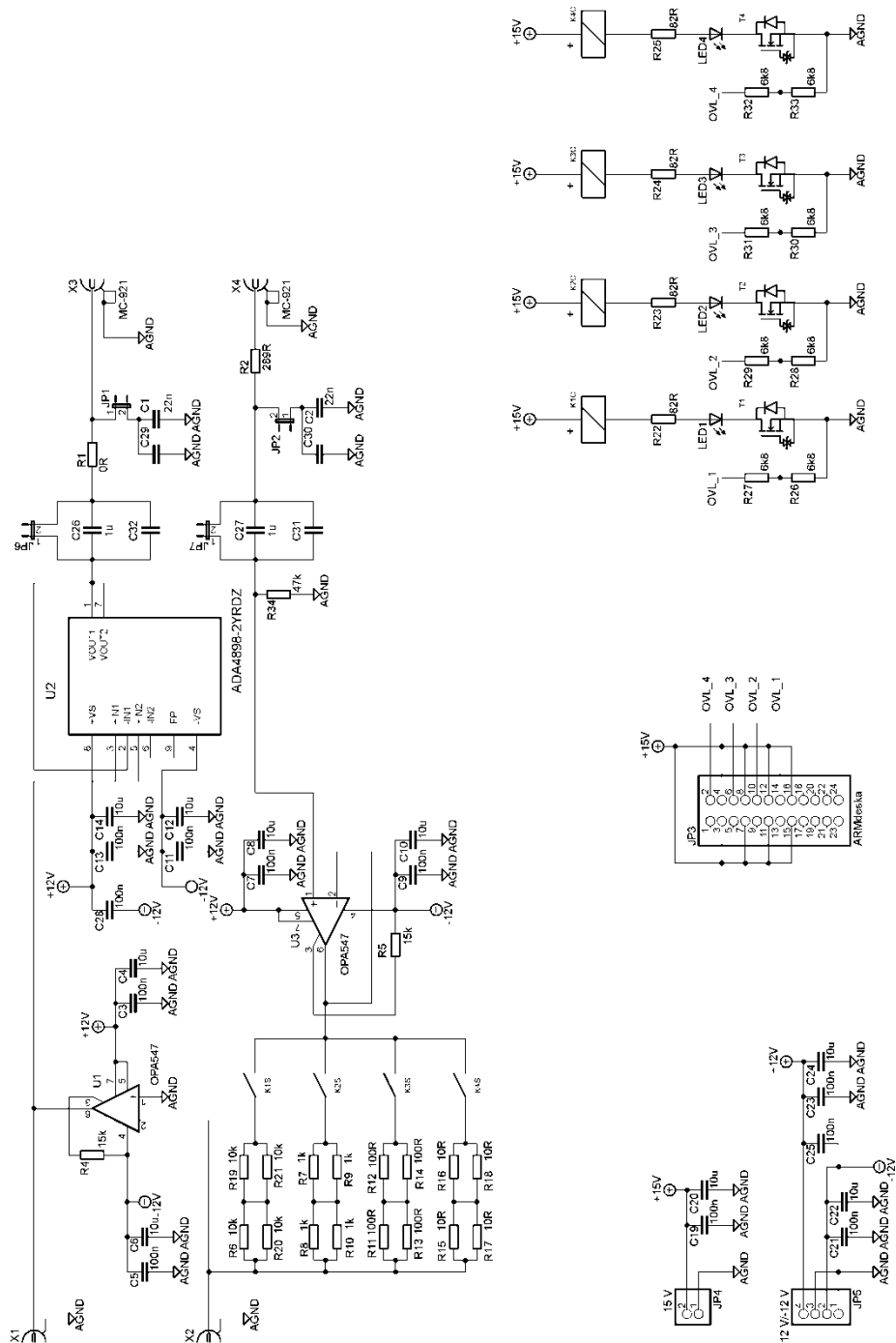
Modul převodníku SI  
část D/A

J. Budasz, J. Necasek

31.12.2013

2/2

## Příloha F Schéma analogového výstupního modulu

Výstupní analogový  
modul SI

J. Budasz, J. Necasek

31.12.2013

1/1



## Příloha G Výpisy skriptů v MATLABu

### Výpis skriptu PorovnaniDiskretizacnichMetod.m

```
close all;      % zavre vsechna okna matlabu
clear all;      % vymaze promenne ve workspace
clc;           % vymaze prikazovy radek
format long;
Fs = 100000;    % vzorkovací frekvence
T = 1/Fs;       % perioda vzorkovani
s = tf('s');    % s jako promenna pro spojite symbolicke vypocty
z = tf('z',T);  % z jako promenna pro diskř. symb. vypocty s T
%% Vektory frekvenci pro vypocty charakteristik
Fvek = logspace(1,4,100); % log. vek. frekvenci 100 bodu od 10 Hz do 10 kHz
Fvek = Fvek(1:91);
Wvek = Fvek*2*pi; % log. vektor odpovídajících kruhových frekvenci v rad/s
%% Definice impedanci
% nutno odkomentovat prislusnou cast, ostatni zakomentovat
%% Cínný odpor R
% W = 1000;
% R = 10000;
% Z = tf(R,1);
% popis = strcat('Čínný odpor R, R = ',num2str(R),' \Omega, W = ',...
%   num2str(W),' \Omega, f_s = ',num2str(Fs/1000),' kHz');
%% Seriový RC člen
% W = 1000;
% C = 1.0000e-05; %
% R = 1000;
% Z = R + 1/(s*C);
% popis = strcat('RC-ser, R = ',num2str(R),' \Omega, C = ',...
%   num2str(C*1000000),' \mu F, W = ',num2str(W),' \Omega, f_s = ',...
%   num2str(Fs/1000),' kHz');
%% Kombinovaný seriový RC člen - delič zatížený kondenzátorem
% W = 1000;
% C = 1.0000e-06; % 1 uF
% R1 = 300;
% R2 = 1000;
% Z = R1 + 1/(1/R2 + s*C);
% popis = strcat('RR dělič se zátěží C, R_1 = ',num2str(R1),...
%   ' \Omega, R_2 = ',num2str(R2),' \Omega, C = ',num2str(C*1000000),...
%   ' \mu F, W = ',num2str(W),' \Omega, f_s = ',num2str(Fs/1000),' kHz');
%% Kombinovaný Seriový RL obvod - delič zatížený cívkou
% W = 1000;
% L = 0.1;
% R1 = 1000;
% R2 = 300;
% Z = R1 + 1/(1/(s*L) + 1/R2);
% popis = strcat('RR dělič se zátěží L, R_1 = ',num2str(R1),...
%   ' \Omega, R_2 = ',num2str(R2),' \Omega, L = ',num2str(L),...
%   ' H, W = ',num2str(W),' \Omega, f_s = ',num2str(Fs/1000),' kHz');
%% Paralelní kombinace dvou seriových RC členu, různá tau
% W = 100;
% R1 = 800;
% C1 = 0.8e-5;
% R2 = 300;
% C2 = 0.3e-6;
% f1 = 1/(2*pi*R1*C1);
% f2 = 1/(2*pi*R2*C2);
% Z = (R1*R2*C1*C2*s^2 + s*(R1*C1 + R2*C2) + 1)/...
%   (C1*C2*(R1+R2)*s^2 + s*(C1+C2));
% popis1 = strcat('Par. komb. 2 RC-ser, R_1 = ',num2str(R1),...
%   ' \Omega, C_1 = ',num2str(C1*1000000),' \mu F, R_2 = ',num2str(R2),...
%   ' \Omega, C_2 = ',num2str(C2*1000000),' \mu F,');
% popis2 = strcat('W = ',num2str(W/1000),' k\Omega, f_s = '...
%   ,num2str(Fs/1000),' kHz');
```



```

% popis = char(popis1,popis2);
%% Seriovy RLC clen
W = 100;
C = 5.0000e-06; %
R = 100;
L = 0.05;
Z = (L*C*s^2 + R*C*s + 1)/(s*C);
f0 = 1/(2*pi*sqrt(L*C));
popis = strcat('RLC-ser, R = ',num2str(R),'\Omega, L = ',num2str(L),...
    ' H, C = ',num2str(C*1000000),'\mu F, f_0 = ',num2str(f0),...
    ' Hz, W = ',num2str(W),'\Omega, f_s = ',num2str(Fs/1000),' kHz');
%% Prepočet na FU
Fu = W/Z; % Fu - spojitý přenos je potřeba implementovat v ARMu jako filtr
%% Adams-Bashforth
[NumFuABR, DenFuABR] = tfdata(Fu); % vektory citatele a jmenovatele
NumFuAB = cell2mat(NumFuABR); % převod z proměnné typu cell na vektor
DenFuAB = cell2mat(DenFuABR);

syms s; % s a z jako symbolické proměnné
syms z;

% vytvoření přenosu v proměnné typu sym, nutné pro
% symbolický výpočet
FuEmulAdBSpoj = poly2sym(NumFuAB,s)/poly2sym(DenFuAB,s);
AdBsubs = (12/T)*(1 - z^-1)/(5 + 8*z^-1 - z^-2); % Adams-Bashford
FuEmulAdBSym = subs(FuEmulAdBSpoj,s,AdBsubs); % substituce
FuEmulAdBSym = simple(sym(FuEmulAdBSym)); % zjednodušení disk. přenosu
[FuEmulAdBSymNum, FuEmulAdBSymDen] = numden(FuEmulAdBSym); % cit. a jmenov.
FuEmulAdBNum = sym2poly(FuEmulAdBSymNum); % převod z proměnné typu sym
FuEmulAdBDen = sym2poly(FuEmulAdBSymDen); % do vektoru

s = tf('s'); % s jako proměnná pro spojitý přenosové funkce
z = tf('z',T); % z jako proměnná pro diskretní přenosové funkce

FuEmulAdB = tf(FuEmulAdBNum,FuEmulAdBDen,T); % vytvoření disk. přenosu
ZEmulAdB = W/FuEmulAdB % Skutečná diskretní impedance

%% Tustin
FuzTust = c2d(Fu,T,'tustin'); % diskretní přenos pomocí Tustin. metody
ZEmulTust = W/FuzTust
%% Pole Mapping
[numFuR, denFuR] = tfdata(Fu); % vektory citatele a jmenovatele
numFu = cell2mat(numFuR); % převod do tvaru matice (vektoru)
numFu = fliplr(numFu); % otočení vektoru koef
denFu = cell2mat(denFuR); % převod do tvaru matice (vektoru)
rootsDenFu = roots(denFu); % výpočet kořenu jmenovatele
denFu = fliplr(denFu); % otočení vektoru koef
denFuRootsCnt = length(rootsDenFu); % počet kořenu jmenovatele
%% Mapování polu přenosu do z-rovin
if(denFuRootsCnt > 0) % pojistka pro případ jednotkového jmenovatele
    rootsDenFuz = zeros(denFuRootsCnt,1); % vektor nových kořenu - v Z rovině
    for a = 1 : denFuRootsCnt
        % mapování všech kořenu z s-oblasti do z-oblasti
        rootsDenFuz(a) = exp(T*rootsDenFu(a));
    end
    % výpočet koeficientu polynomu z kořenu jmenovatele disk. přenosu
    denFuz = poly(rootsDenFuz);
    Avek = denFuz'; % transpozice koeficientu jmen. do sloupc. vek. Avek
    C = zeros(denFuRootsCnt + 1,denFuRootsCnt + 1); % příprava matic
    D = zeros(denFuRootsCnt + 1,denFuRootsCnt + 1);
    H = zeros(denFuRootsCnt + 1,denFuRootsCnt + 1);
    for a = 1 : denFuRootsCnt + 1 % procházení matic a - index radku
        for b = 1 : denFuRootsCnt + 1 % b - index sloupce
            C(a,b) = ((-T*(b-1))^(a-1))/factorial(a-1); % prvek matic C a V
            if(a - b) > -1 % pokud jsme na diagonale a pod ní
                D(a,b) = denFu(a-b+1); % zápis příslušných koef. do matic
            end
        end
    end

```



```

        H(a,b) = numFu(a-b+1); % D a H
    end
end
end
V = C; % vytvoreni matice V, je stejná jako C

Bvek = inv(V)*H*inv(D)*C*Avek;% Hlavní výpočet sloupčového vektoru Bvek
% Bvek je vektor koeficientu citatele
% diskretního přenosu

FuEmulPoleMap = tf(Bvek',denFuz,T); % vytvoreni disk. přenosu z vektoru
ZEmulPoleMap = W/FuEmulPoleMap % skutecná implementovaná impedance
else % varování pokud metodu nelze použít
    disp('Přenos nelze diskretizovat metodou mapování polu, Fu bez polu');
end
%% Výpočet frekvenčních průbehů pro různé metody
Zideal = Z; % Ideální průběh
[magIdealR,phaseIdealR,wout] = bode(Zideal,Wvek); % ideal - spojitý
[magEmulTustR,phaseEmulTustR,wout1] = bode(ZEmulTust,Wvek); % Tustinova m.
[magEmulAdBR,phaseEmulAdBR,wout2] = bode(ZEmulAdB,Wvek); % Adams-Bashforth
% převedení všech vektorů do správného tvaru
magIdeal = reshape(magIdealR,length(Wvek),1);
phaseIdeal = reshape(phaseIdealR,length(Wvek),1);
magEmulTust = reshape(magEmulTustR,length(Wvek),1);
phaseEmulTust = reshape(phaseEmulTustR,length(Wvek),1);
magEmulAdB = reshape(magEmulAdBR,length(Wvek),1);
phaseEmulAdB = reshape(phaseEmulAdBR,length(Wvek),1);
if(denFuRootsCnt > 0) % pokud bylo možné použít Pole Mapping
    % výpočet frek. průbehů emulované Pole Mapping Z v bodech Wvek
    [magEmulPoleMapR,phaseEmulPoleMapR,wout3] = bode(ZEmulAdB,Wvek);
    magEmulPoleMap = reshape(magEmulPoleMapR,length(Wvek),1);
    phaseEmulPoleMap = reshape(phaseEmulPoleMapR,length(Wvek),1);
end
%% Zobrazení
figure; % nové okno pro graf
semilogx(Fvek,phaseIdeal,'red',... % vykreslí 3 fázové charak.
Fvek,phaseEmulTust,'blue',... % , osa x logaritmičtý
Fvek,phaseEmulAdB,'green','LineWidth',2);

hold on;
if(denFuRootsCnt > 0) % pokud bylo možné použít Pole Mapping, vykreslí ji
    semilogx(Fvek,phaseEmulPoleMap,'black',... % vykreslí fázi Pole-mapping
'LineWidth',2.5,'LineStyle','- -');
end
set(gca,'FontSize',15); % nastavení velikosti čísel grafu
xlabel('Frekvence [Hz]','fontsize',15); % popiska osy x
ylabel('Fáze [°]','fontsize',15); % popiska osy y
title({'Fáze impedance Z, zadaná versus emulovaná';popis},...
'fontsize',15); % nadpis grafu
grid on; % zapne mřížku
legend('Spojité Z','Bilineární disk. Z',...
'Ad-Bash. disk. Z','Map. půlů disk. Z',...
'Location','Best'); % legenda grafu
figure % nové okno pro graf
loglog(Fvek,(magIdeal),'red',...% vykreslí 3 amplitudové charak.,
Fvek,(magEmulTust),'blue',... % osy logaritmičtý
Fvek,(magEmulAdB),'green','LineWidth',2);

hold on;
if(denFuRootsCnt > 0) % pokud bylo možné použít Pole Mapping, vykreslí ji
    semilogx(Fvek,magEmulPoleMap,'black','LineWidth',4,'LineStyle','-');
end
set(gca,'FontSize',15); % nastavení velikosti čísel grafu
xlabel('Frekvence [Hz]','fontsize',15); % popiska osy x
ylabel('Absolutní hodnota [Ω]','fontsize',15); % popiska osy y
title({'Absolutní hodnota impedance Z, zadaná versus emulovaná';popis},...
'fontsize',15); % nadpis grafu

```





```

legend( 'Spojitá Z','Bilineární disk. Z',...
        'Ad-Bash. disk. Z','Map. pólů disk. Z',...
        'Location','Best'); % legenda grafu
grid on; % zapne mřížku
%% Uložení průběhu
save('ZadejNazev.mat','Fvek','magIdeal',...
     'magEmulTust','magEmulAdB','magEmulPoleMap','phaseIdeal',...
     'phaseEmulTust','phaseEmulAdB','phaseEmulPoleMap');

```

**Výpis skriptu ZapisDatRS232.m**

```

clc;
format long;
% Kod zápise koeficienty přenosu uložene ve vektorech
% FuFilterNum2Send a FuFilterDen2Send. Dale zapisuje
% index sepnutého rele výstupu, z proměnné indexWout.
% Pokud jedna z těchto proměnných neexistuje, kod
% je zastaven a neprovede vůbec nic. Potřebná data
% čte z pracovního prostoru globálních dat MatLabu
% Správný formát dat:
% FuFilterNum2Send - oboje typ single (float32)
% FuFilterDen2Send - oboje řádkový vektor délky 10
% indexWout - typ uint8, obsahuje čísla mezi 1 a 4
% Pomocné proměnné markery existence proměnných
NumVekExists = exist('FuFilterNum2Send','var'); % Testuje existenci
DenVekExists = exist('FuFilterDen2Send','var'); % proměnné pod daným
IndxWoutExists = exist('indexWout','var'); % nazvem
if(or(or((NumVekExists == 0),(DenVekExists == 0)),(IndxWoutExists == 0)))
    error('Nebyly nalezeny potřebné proměnné pro zápis.');
```

**end**

**try**

```

    % Zadání parametru přenosu
    serialPort = serial('COM4'); % Jméno portu
    serialPort.BaudRate = 921600; % Rychlost
    serialPort.DataBits = 8; % Počet dat. bitu
    serialPort.FlowControl = 'hardware'; % Rízení přenosu - HW
    serialPort.Parity = 'none'; % Bez parity
    serialPort.StopBits = 1; % 1 stop bit
    serialPort.ByteOrder = 'littleEndian'; % Pro posílání delších dat,
    % nespojující s rozložením dat
    % v RAM ARMu, lepší volba pro přenos

    serialPort.RequestToSend = 'on'; % RTS on
    serialPort.Timeout = 10; % Časový limit pro příjem i odeslání
    % pak chyba

    serialPort.OutputBufferSize = 100; % Vstupní a výstupní buffer
    serialPort.InputBufferSize = 100;
    fopen(serialPort); % Otevře port
    temp = uint8(0); % Dočasná proměnná k odeslání, jeden byte
    temp = hex2dec('BF'); % Kod pro zápis dat o přenosu z ARMu
    fwrite(serialPort,temp,'uint8'); % Zápis do portu, tím i odeslání
    fwrite(serialPort,indexWout,'uint8'); % Zápis do portu, tím i odeslání
    for a = 1:10
        % Zápis do portu, tím i odeslání
        fwrite(serialPort,FuFilterDen2Send(a),'float32');
    end
    for a = 1:10
        % Zápis do portu, tím i odeslání
        fwrite(serialPort,FuFilterNum2Send(a),'float32');
    end
    catch % Při zachycení chyby v předchozím kodu
        disp('Chyba při odeslání dat'); % Varování při chybě
    end
    sExistFlag = exist('serialPort'); % Zjistuje, zda proměnná portu existuje
    if(sExistFlag == 1)
        fclose(serialPort); % Pokud ano, zavře její a vymaže
        delete(serialPort)
    end

```



```
clear serialPort
end
```

### Výpis skriptu CteniDatRS232.m

```
clc;
format long;
% Kod cte aktualny koeficienty prenosu v ARMu
% a nastaveni prenosu. Na zadany seriový port
% odesle jeden byte - kod pro vyceteni a ceká na data.
% Prijmuta data zobrazí.
try
    % Zadání parametru prenosu
    serialPort = serial('COM4'); % Jmeno portu
    serialPort.BaudRate = 921600; % Rychlost
    serialPort.DataBits = 8; % Pocet dat. bitu
    serialPort.FlowControl = 'hardware'; % Rizení prenosu - HW
    serialPort.Parity = 'none'; % Bez parity
    serialPort.StopBits = 1; % 1 stop bit
    serialPort.ByteOrder = 'littleEndian'; % Pro posilání delších dat,
    % nesouvisí s rozložením dat
    % v RAM ARMu, lepší volba pro prenos
    serialPort.RequestToSend = 'on'; % RTS on
    serialPort.Timeout = 4; % Casový limit pro příjem i odeslání
    % pak chyba
    % Vstupní a výstupní buffer
    serialPort.OutputBufferSize = 1;
    serialPort.InputBufferSize = 100;
    fopen(serialPort); % Otevře port
    temp = uint8(0); % Dočasná proměnná k odeslání, jeden byte
    temp = hex2dec('CD'); % Kod pro vyceteni dat o prenosu z ARMu
    fwrite(serialPort,temp,'uint8'); % Zápis do portu, tím i odeslání
catch % Při zachycení chyby v předchozím kodu
    disp('Chyba při odeslání dat'); % Varování při chybě
end
try
    rcvdDataIndxW = fread(serialPort,1,'uint8'); % Příjem jednoho byte dat
    rcvdDataBvek = fread(serialPort,10,'float32'); % Příjem 10 koef. float
    rcvdDataAvek = fread(serialPort,10,'float32'); % Příjem 10 koef. float
    disp(rcvdDataIndxW);
    disp(rcvdDataBvek); % Zobrazení přijatých dat
    disp(rcvdDataAvek);
    fclose(serialPort); % Uzavření portu
    delete(serialPort) % Vymazání proměnné port
    clear serialPort;
catch % Při zachycení chyby v předchozím kodu
    disp('Chyba při příjmu dat') % Varování při chybě
    sExistFlag = exist('serialPort'); % Zjistuje, zda proměnná portu existuje
    if(sExistFlag == 1)
        fclose(serialPort); % Pokud ano, zavře jej a vymaze
        delete(serialPort)
        clear serialPort
    end
end
end
```

### Výpis skriptu UlozeniDatKOdeslani.m

```
% Kod uloží proměnné k odeslání
% FuFilterNum2Send, FuFilterDen2Send a indexWout
% z pracovního prostoru do souboru s příslušným názvem.
save('ZadejNazev.mat','FuFilterDen2Send','FuFilterNum2Send','indexWout');
```

### Výpis skriptu NacteniDatKOdeslani.m

```
% Kod načte proměnné k odeslání, tj.
% FuFilterNum2Send, FuFilterDen2Send a indexWout
% z příslušného souboru a uloží je do pracovního prostoru.
load('ZadejNazev.mat');
```



## Výpis skriptu ZakladniImpedance.m

```

close all;      % zavre vsechna okna matlabu
clear all;      % vymaze promenne ve workspace
clc;           % vymaze prikazovy radek
format long;
Fs = 100000;    % vzorkovaci frekvence
T = 1/Fs;       % perioda vzorkovani
s = tf('s');    % s jako promenna pro spojite symbolicke vypocty
z = tf('z',T);  % z jako promenna pro diskr symb. vypocty s T
%% Vektory frekvenci pro vypocty charakteristik
Fvek = logspace(1,4,100); % log. vek. frekvenci 100 bodu od 10 Hz do 10 kHz
Fvek = Fvek(1:91);
Wvek = Fvek*2*pi; % log. vektor odpovídajících kruhových frekvenci v rad/s
%% Definice impedanci
% nutno odkomentovat prislusnou cast, ostatni zakomentovat
%% Cinný odpor R
% W = 1000;
% R = 10000;
% Z = tf(R,1);
% popis = strcat('Činný odpor R, R = ',num2str(R),' \Omega, W = ',...
%   num2str(W),' \Omega, f_s = ',num2str(Fs/1000),' kHz');
%% Seriovy RC clen
% W = 1000;
% C = 1.0000e-05; %
% R = 1000;
% Z = R + 1/(s*C);
% popis = strcat('RC-ser, R = ',num2str(R),' \Omega, C = ',...
%   num2str(C*1000000),' \mu F, W = ',num2str(W),' \Omega, f_s = ',...
%   num2str(Fs/1000),' kHz');
%   % Testovani nestability
%   if((R*C) <= 0) % pokud je nestabilni
%       error('Zadane hodnoty vedou na nestabilni prenos Fu!');
%   end
%% Kombinovany seriovy RC clen - delic zatizeny kondenzátorem
% W = 1000;
% C = 1.0000e-06; % 1 uF
% R1 = 300;
% R2 = 1000;
% Z = R1 + 1/(1/R2 + s*C);
% popis = strcat('RR dělič se zátěží C, R_1 = ',num2str(R1),...
%   ' \Omega, R_2 = ',num2str(R2),' \Omega, C = ',num2str(C*1000000),...
%   ' \mu F, W = ',num2str(W),' \Omega, f_s = ',num2str(Fs/1000),' kHz');
%   % Testovani nestability
%   if((R1 + R2)/(C*R1*R2)) <= 0) % pokud je nestabilni
%       error('Zadane hodnoty vedou na nestabilni prenos Fu!');
%   end
%% Kombinovany Seriovy RL obvod - delic zatizeny cívku
% W = 1000;
% L = 0.1;
% R1 = 1000;
% R2 = 300;
% Z = R1 + 1/(1/(s*L) + 1/R2);
% popis = strcat('RR dělič se zátěží L, R_1 = ',num2str(R1),...
%   ' \Omega, R_2 = ',num2str(R2),' \Omega, L = ',num2str(L),...
%   ' H, W = ',num2str(W),' \Omega, f_s = ',num2str(Fs/1000),' kHz');
%   % Testovani nestability
%   if((R1*R2/(L*(R1 + R2))) <= 0) % pokud je nestabilni
%       error('Zadane hodnoty vedou na nestabilni prenos Fu!');
%   end
%% Paralelní kombinace dvou seriových RC clenu, ruzna tau
% W = 100;
% R1 = 800;
% C1 = 0.8e-5;
% R2 = 300;

```



```

% C2 = 0.3e-6;
% f1 = 1/(2*pi*R1*C1);
% f2 = 1/(2*pi*R2*C2);
% Z = (R1*R2*C1*C2*s^2 + s*(R1*C1 + R2*C2) + 1)/...
% (C1*C2*(R1+R2)*s^2 + s*(C1+C2));
% popis1 = strcat('Par. komb. 2 RC-ser, R_1 = ',num2str(R1),...
% '\Omega, C_1 = ',num2str(C1*1000000),' \muF, R_2 = ',num2str(R2),...
% '\Omega, C_2 = ',num2str(C2*1000000),' \muF,');
% popis2 = strcat('W = ',num2str(W/1000),' k\Omega, f_s = '...
% ',num2str(Fs/1000),' kHz');
% popis = char(popis1,popis2);
% % Testovani nestability
% if((R1*C1) <= 0 || (R2*C2) <= 0) % pokud je nestabilni
%     error('Zadane hodnoty vedou na nestabilni prenos Fu!');
% end
%% Seriovy RLC clen
W = 100;
C = 5.0000e-06; %
R = 100;
L = 0.05;
Z = (L*C*s^2 + R*C*s + 1)/(s*C);
f0 = 1/(2*pi*sqrt(L*C));
popis = strcat('RLC-ser, R = ',num2str(R),' \Omega, L = ',num2str(L),...
' H, C = ',num2str(C*1000000),' \muF, f_0 = ',num2str(f0),...
' Hz, W = ',num2str(W),' \Omega, f_s = ',num2str(Fs/1000),' kHz');
% Testovani nestability
condStab1 = ((-R*C + j*sqrt(4*L*C - R^2*C^2))/(2*L*C)); % koeficienty
condStab2 = ((-R*C - j*sqrt(4*L*C - R^2*C^2))/(2*L*C)); % pro testovani
if(real(condStab1)>= 0 || real(condStab2) >= 0) % pokud je nestabilni
    error('Zadane hodnoty vedou na nestabilni prenos Fu!');
end
%% Prepocet na Fu
Fu = W/Z; % Fu - spojity prenos je potreba impementovat v ARMu jako filtr
%% Pridavne filtry
Rf1 = 289; % hodnoty soucastek filtru
Rf2 = 47000;
Cf = 2.2e-8;
Ffilter = Rf2/(Rf1*Rf2*Cf*s + Rf1 + Rf2); % Prenos antialiasing filtru
%% Tustinova diskretizace
FuzTust = c2d(Fu,T,'tustin') ; % idealni diskretni prenos pomoci Tustina
DSPDelay = 1; % zpozdeni realneho vypoctu
FuzDelayedTust = FuzTust*(z^(-DSPDelay)); % realny prenos se zpozdenim DSP
ZEmulTust = W/FuzDelayedTust; % Diskretni model real. impedance
[num,den,Ts] = tfdata(FuzTust); % Vycteni koef. Fu pro pripravu prenosu
FuFilterDen = cell2mat(den); % Prevedeni datovych formatu
FuFilterNum = cell2mat(num);
N = length(FuFilterDen); % Zjisteni radu jmenovatele N
M = length(FuFilterNum); % a citatele M
% figure; % jen pro ladeni
% pzplot(FuzTust); % pomocne zobrazeni polu a nul prenosu FuzTust
%% Vypocet prubehu
Zideal = W/FuzTust; % Idealni prubeh
[magIdealR,phaseIdealR,wout] = bode(Zideal,Wvek); % frek. prubehu idealni Z
[magEmulTustR,phaseEmulTustR,wout1] = bode(ZEmulTust,Wvek); % synt. imped.
[magFilterR,phaseFilterR,wout2] = bode(Ffilter,Wvek); % frekvencni prubeh
% pridavnych filtru
% prevedeni vseh vektoru do spravneho tvaru
magIdeal = reshape(magIdealR,length(Wvek),1);
phaseIdeal = reshape(phaseIdealR,length(Wvek),1);
magEmulTust = reshape(magEmulTustR,length(Wvek),1);
phaseEmulTust = reshape(phaseEmulTustR,length(Wvek),1);
magFilter = reshape(magFilterR,length(Wvek),1);
phaseFilter = reshape(phaseFilterR,length(Wvek),1);
% pripocteni vlivu pridavnych filtru na amplitudu a fazi
magEmul = magEmulTust./magFilter;

```



```

phaseEmul = phaseEmulTust - phaseFilter;
%% Zobrazení průběhu
figure; % nové okno pro graf
semilogx(Fvek, phaseIdeal, 'red', ... % vykreslí 2 fázové charak.
          Fvek, phaseEmul, 'blue', ... % osa x logaritmicky
          'LineWidth', 2);
set(gca, 'FontSize', 15); % nastavení velikosti čísel grafu
xlabel('Frekvence [Hz]', 'fontSize', 15); % popiska osy x
ylabel('Fáze [°]', 'fontSize', 15); % popiska osy y
title({'Fáze impedance Z'; popis}, ...
      'fontSize', 15); % nadpis grafu
grid on; % zapne mřížku
legend('Spojitá Z', 'Syntetizovaná Z', ...
       'Location', 'Best'); % legenda grafu
figure; % nové okno pro graf
loglog(Fvek, magIdeal, 'red', ... % vykreslí 2 amplitudové charak.,
        Fvek, magEmul, 'blue', ... % osy logaritmicky
        'LineWidth', 2);
hold on;
set(gca, 'FontSize', 15); % nastavení velikosti čísel grafu
xlabel('Frekvence [Hz]', 'fontSize', 15); % popiska osy x
ylabel('Absolutní hodnota [Ω]', 'fontSize', 15); % popiska osy y
title({'Absolutní hodnota impedance Z'; popis}, ...
      'fontSize', 15); % nadpis grafu
legend('Spojitá Z', 'Syntetizovaná Z', ...
       'Location', 'Best'); % legenda grafu
grid on; % zapne mřížku
%% Vytvoření promenných pro odeslání do ARMu
% Vytvoření dvou řádkových vektorů, dlouhých 10.
% Jeden pro čitatele, jeden pro jmenovatele.
% Pro požadovanou délku doplněno nulami.
FuFilterDen2Send = single([FuFilterDen ...
                           zeros(1, 10-length(FuFilterDen))]');
FuFilterNum2Send = single([zeros(1, N-M) FuFilterNum ...
                           zeros(1, 10-length(FuFilterNum)-(N-M))]');
% Přepočítání hodnoty odporu W na index rele výstupu.
% 10 Ohm odpovídá 1
% 100 Ohm 2
% 1000 Ohm 3
% 10000 Ohm 4
indexWout = uint8(4);
switch W
case 10
    indexWout = uint8(1);
case 100
    indexWout = uint8(2);
case 1000
    indexWout = uint8(3);
case 10000
    indexWout = uint8(4);
otherwise
    indexWout = uint8(4);
end

```



### Výpis skriptu FIRcritFvek.m

```
function J = FIRcritFvek(x,M,FdVek,K,Dmag,Dfi,PsiMtx,FiMtx)
% Vstupni parametry
% x = [b0 b1 ..... bM-1 bM], delka M+1
% M rad citatele
% FdVek vektor digitalnich frekvenci, ve kterych se pocita charak.,
% Fd je v intervalu 0 az pi/2
% K pocet bodu frekvencni charakteristiky, ve kterych probiha vypocet
% Dmag = [Dmag0 Dmag1 ..... DmagK-1] vektor pozadovane magnitudy, delka K
% Dfi = [Dfi0 Dfi1 ..... DfiK-1] vektor pozadovane faze, delka K
% PsiMtx vahy magnitudy, ctvercova diagonalni matice, rozmer K
% FiMtx vahy faze, ctvercova diagonalni matice, rozmer K
H = zeros(1,K);
for k = 0 : 1 : K-1 %prochazi vsechny frekvence
    for m = 0 : 1 : M %prochazi vsechny clen citatele
        H(1,k+1) = H(1,k+1) + x(m+1)*exp(-1j*FdVek(k+1)*m); % bod charak
    end % pro jednu f.
end
Hmag = abs(H); % vraci vektor abs hodnoty - mag
Hfi = angle(H); % vraci vektor faze
dHmag = Hmag - Dmag; % vektor rozdilu amplitudy
dHfi = Hfi - Dfi; % vektor rozdilu faze
J = dHmag*PsiMtx*dHmag' + dHfi*FiMtx*dHfi'; % hodnota kriteria
```

### Výpis skriptu IIRcritFvek.m

```
function J = IIRcritFvek(x,M,N,FdVek,K,Dmag,Dfi,PsiMtx,FiMtx)
% Vstupni parametry
% x = [b0 b1 ..... bM-1 bM a1 a2 ..... aN], delka M+1 + N
% a0 = 1 vzdy!!! a0 neni obsazen ve vektoru x!
% M rad citatele
% N rad jmenovatele
% FdVek vektor digitalnich frekvenci, ve kterych se pocita charak.,
% Fd je v intervalu 0 az pi
% K pocet bodu frekvencni charakteristiky, ve kterych probiha vypoce
% Dmag = [Dmag0 Dmag1 ..... DmagK-1] vektor pozadovane magnitudy, delka K
% Dfi = [Dfi0 Dfi1 ..... DfiK-1] vektor pozadovane faze, delka K
% PsiMtx vahy magnitudy, ctvercova diagonalni matice, rozmer K
% FiMtx vahy faze, ctvercova diagonalni matice, rozmer K
H = zeros(1,K);
for k = 0 : 1 : K-1 %prochazi vsechny frekvence
    for m = 0 : 1 : M %prochazi vsechny clen citatele
        H(1,k+1) = H(1,k+1) + x(m+1)*exp(-1j*FdVek(k+1)*m); % citatel
    end
    denom = 1;
    for n = 1 : 1 : N %prochazi vsechny clen jmenovatele
        denom = denom + x(M+1+n)*exp(-1j*FdVek(k+1)*n); % jmenovatel
    end
    if(denom ~= 0)
        H(1,k+1) = H(1,k+1)/denom; % bod charak. pro jednu frekvenci
    else %pokud se deli nulou
        disp('Chyba ve vypoctu hodnot charakteristiky');
    end
end
Hmag = abs(H); % vraci vektor abs hodnoty - mag
Hfi = angle(H); % vraci vektor faze
dHmag = Hmag - Dmag; % rozdilove vektory
dHfi = Hfi - Dfi;
J = dHmag*PsiMtx*dHmag' + dHfi*FiMtx*dHfi'; %vysledek - kriterium
```



### Výpis skriptu IIRRootsConstrain.m

```
function [c,ceq] = IIRRootsConstrain(x,M,N)
% Vstupni parametry
% x = [b0 b1 ..... bM-1 bM a1 a2 .....aN], delka M+1 + N
% a0 = 1 vzdy!!! a0 neni obsazen ve vektoru x!
% M rad citatele
% N rad jmenovatele
% vraci vektor c, kladny v pripade nestabilnich koeficientu jmenovatele
% zaporny pokud jsou stabilni
A = [1 x(M+2:M+N+1)]; % doplneni na kompletni vektor jmenovatele
AbsrootsA = abs((roots(A))'); % vektor absolutnich hodnot korenu jmen.
c = AbsrootsA - 0.999.*ones(1,length(AbsrootsA)); % kriterialni vektor
ceq = zeros(1,length(x)); % vektor jen pro formu, musi tu byt
[c,ceq]; % vraceni vysledku
```

### Výpis skriptu RandInitPolyDesign.m

```
function KoefVek = RandInitPolyDesign(N)
%InitPolyDesign Navrhuje pocatecni bod (polynom radu N) pro optimalizaci
% Navrhne polynom radu N
% Vsechny koreny polynomu lezi uprostred jednotkove kruznice
% Vysledne koeficienty polynomu jsou ve vystupnim vektoru
% KoefVek = [1 ..... ]
singleRootCnt = rem(N,2); % Pocet samotnych real. korenu
conjCouplesCnt = (N - singleRootCnt)/2; % Pocet dvojic komplex. sdruz kor.
rootsVek = zeros(1,singleRootCnt + 2*conjCouplesCnt); % Celkovy pocet kor.
if (singleRootCnt > 0) % Pokud existuje nejaky realny koren
    rootsVek(1) = random('beta',1,1); % vygeneruje ho
end
for a = 0 : 1 : conjCouplesCnt-1 % Pro vsechny dvojice komplex sdruz korenu
    tempAbs = random('beta',1,1); % Navrhne abs hodnotu
    tempArg = random('beta',1,1)*pi; % Navrhne argument
    rootsVek(1,singleRootCnt+1+2*a) = tempAbs*exp(1j*tempArg); % Slozi do
    rootsVek(1,singleRootCnt+1+2*a+1) = tempAbs*exp(-1j*tempArg); % dvojice
    % komplexne sdruzenych korenu
end
KoefVek = poly(rootsVek); % Vytvori koeficienty polynomu z korenu a vrati
end
```

### Výpis skriptu RandInitPolyDesignNoConstrain.m

```
function KoefVek = RandInitPolyDesignNoConstrain(N)
%InitPolyDesign Navrhuje pocatecni bod (polynom radu N) pro optimalizaci
% Navrhne polynom radu N
% Vsechny koreny polynomu lezi uprostred kruznice o polomeru 10
% Vysledne koeficienty polynomu jsou ve vystupnim vektoru
% KoefVek = [1 ..... ]
singleRootCnt = rem(N,2); % Pocet samotnych real. korenu
conjCouplesCnt = (N - singleRootCnt)/2; % Pocet dvojic komplex. sdruz korenu
rootsVek = zeros(1,singleRootCnt + 2*conjCouplesCnt); % Celkovy pocet korenu
if (singleRootCnt > 0) % Pokud existuje nejaky realny koren
    rootsVek(1) = random('beta',1,1); % vygeneruje ho
end
for a = 0 : 1 : conjCouplesCnt-1 % Pro vsechny dvojice komplex sdruz korenu
    tempAbs = 10*random('beta',1,1); % Navrhne abs hodnotu
    tempArg = random('beta',1,1)*pi; % Navrhne argument
    rootsVek(1,singleRootCnt+1+2*a) = tempAbs*exp(1j*tempArg); % Slozi do
    rootsVek(1,singleRootCnt+1+2*a+1) = tempAbs*exp(-1j*tempArg); % dvojice
    % komplexne sdruzenych korenu
end
KoefVek = poly(rootsVek); % Vytvori koeficienty polynomu z korenu a vrati
end
```



## Výpis skriptu NavrhFiltruFIR.m

```

close all;
clear all;
clc;
%% Nastavení základních veličin
W = 100; % W odpor určující rozsah absolutní hodnoty
Fs = 100000; % Vzorkovací frekvence a perioda
T = 1/Fs;
z = tf('z',T); % z jako proměnná pro diskretní přenosové funkce
s = tf('s'); % s jako proměnná pro spojité přenosové funkce
MarkDataRead = 0; % Marker, data načtena ze souboru, inicializace
%% Zadání radu filtru
M = 3; % Rad filtru
%% Načtení dat z tabulkového souboru
roughData = xlsread('data.xlsx'); % čtení z excel souboru
% Vyčtení dat - závisle na struktuře souboru, v případě
% potřeby nutno upravit
FvekData = roughData(:,1); % Vyčtení sloupce, 1. sloupec frekvence
Phase0 = roughData(:,2); % 2. sloupec fáze 0
Phase1 = roughData(:,3); % 3. sloupec fáze 1
Mag1 = roughData(:,4); % 4. sloupec Magnituda 1
Mag0 = roughData(:,5); % 5. sloupec Magnituda 0
ZMagRefData = 100.*Mag0; % Data brána jako zadání optimalizační
ZPhaseRefData = Phase0; % ulohy
FuMagRefData = (W./ZMagRefData)'; % Prepočet průběhu fáze a amplitudy Z
FuPhaseRefDegData = ((-1).*ZPhaseRefData)'; % na průběh filtru Fu
FuPhaseRefRadData = (pi/180).*FuPhaseRefDegData; % Prepočet fáze na radiány
MarkDataRead = 1; % Marker, že data byla načtena ze souboru
%% Parametry frekvencní charakteristiky
FrqMin = 10; % Nejnižší uvažovaná frekvence
FrqIntMin = 10; % Dolní mez oblasti zájmu ve frek., defaultní hodnota
FrqIntMax = 5000; % Horní mez oblasti zájmu ve frek., defaultní hodnota
FrqMax = Fs/2; % Nejvyšší uvažovaná frekvence je Nyquistova
if(MarkDataRead == 1) % Pokud byla data úspěšně načtena, jinak defaultní
    FrqIntMin = FvekData(1);
    FrqIntMax = FvekData(length(FvekData));
end
FrqStepsCntLow = 50; % Počet kroku ve frekvenci pod oblasti zájmu
FrqStepsCntInt = 500; % Počet kroku ve frekvenci v oblasti zájmu
FrqStepsCntUpp = 100; % Počet kroku ve frekvenci nad oblasti zájmu
MinIndexInt = FrqStepsCntLow; % Pomocné indexy pro zobrazování
MaxIndexInt = FrqStepsCntLow + FrqStepsCntInt - 1;
% Dilci vektory frekvence pro jednotlivé části
FvekLow = linspace(FrqMin,FrqIntMin,FrqStepsCntLow);
FvekInt = linspace(FrqIntMin,FrqIntMax,FrqStepsCntInt);
FvekUpp = linspace(FrqIntMax,FrqMax,FrqStepsCntUpp);
% Celkový vektor frekvencí [Hz]
Fvek = [FvekLow(1:FrqStepsCntLow-1) FvekInt FvekUpp(2:FrqStepsCntUpp)];
Wvek = 2*pi.*Fvek; % Celkový vektor kruh frekvencí [rad.s-1]
Fdvek = Wvek./Fs; % Vektor digitálních frekvencí [1]
K = length(Fvek); % Počet bodů frek. charakteristiky
%% Nastavení parametru optimalizace - počáteční bod a váhové matice
x0rough = load('x0storeFIRpok.us.mat'); % Načte uložené koeficienty filtru
x0 = x0rough.x; % použije se jako výchozí bod optimalizace
% Navrhne koeficienty citatele náhodně
tempVekNum = RandInitPolyDesignNoConstrain(M);
if(length(x0) ~= M+1) % pokud načtenou hodnotu ze souboru nelze použít
    disp('Nesouhlasí délka x0 a x, nutno vygenerovat x0!'); % upozornit
    x0 = tempVekNum; % a použije náhodně vygenerovaný
end
%% Následující řádek zakomentovat pokud se pro x0 používá načtená hodnota
% ze souboru; pokud není je pro x0 použita náhodná hodnota
x0 = tempVekNum; %
%%

```





```

PsiMtx = eye(K); % Vahova matice pro optimalizaci, PsiMtx pro magnitudu
FiMtx = eye(K); % Vahova matice pro optimalizaci, FiMtx pro fazi
%% Linearni koeficienty
alfa = 2000; % Hodnoty pro tvorbu vah, alfa pro magnitudu
beta = 1000; % beta pro fazi
% Nasledujici forcyklus vytvori vahy ve tvaru klesajici lin
% zavislosti v oblasti zajmu, jinde je hodnota rovna 1
for a = MinIndexInt : MaxIndexInt % Zdurazneni hodnot ve vah. maticich
    PsiMtx(a,a) = ((alfa - 1)/...
        (MinIndexInt - MaxIndexInt))*(a - MinIndexInt) + 2*alfa;
    FiMtx(a,a) = ((beta - 1)/...
        (MinIndexInt - MaxIndexInt))*(a - MinIndexInt) + 2*beta;
end
% UkoefVek = (diag(PsiMtx)); % Pomocne vektory diagonalnich prvku matic
% WkoefVek = (diag(FiMtx)); % PsiMtx a FiMtx, slouzi jen pro zobrazeni
% figure; % Pouze pomocne zobrazeni prubehu vahovych koeficientu
% plot(Fvek,UkoefVek,'red',Fvek,WkoefVek,'blue');
%% Okno
% MinIndexWindow = find(Fvek > 200,1); % pocatek okna, index dle frekvence
% MaxIndexWindow = find(Fvek > 5000,1); % konec okna
% WindowLength = MaxIndexWindow - MinIndexWindow; % delka okna
% WindowVek = bohmanwin(WindowLength); % volba okna, mozno dat jine dle
% % WindowVek = rectwin(WindowLength); % napovedy
% alfa = 10000; % Hodnoty pro tvorbu vah, alfa pro magnitudu
% beta = 10000; % beta pro fazi
% % Nasledujici forcyklus vytvori vahy ve dle okenkovaciho
% % vektoru v rozsahu okna zadaniem vyse
% for a = MinIndexWindow : MaxIndexWindow - 1
%     PsiMtx(a,a) = PsiMtx(a,a) + WindowVek(a-MinIndexWindow+1)*alfa;
%     FiMtx(a,a) = FiMtx(a,a) + WindowVek(a-MinIndexWindow+1)*beta;
% end
% % UkoefVek = (diag(PsiMtx)); % Pomocne vektory diagonalnich prvku matic
% % WkoefVek = (diag(FiMtx)); % PsiMtx a FiMtx, slouzi jen pro zobrazeni
% % figure; % Pouze pomocne zobrazeni prubehu vahovych koeficientu
% % semilogx(Fvek,UkoefVek,'red',Fvek,WkoefVek,'blue');
%% Interpolace dat splajnem pro moznost ruzneho kroku dat a vypoctu
FuPhaseRefDataInterp = spline(FvekData,FuPhaseRefRadData,FvekInt); % faze
FuMagRefDataInterp = spline(FvekData,FuMagRefData,FvekInt); % amplituda
%Pomocne zobrazeni dat po interpolaci
% figure;
% semilogx(FvekData,FuPhaseRefRadData,...
%     'red',FvekInt,FuPhaseRefDataInterp,'x');
% title('Phase');
% figure;
% semilogx(FvekData,(FuMagRefData),
%     'red',FvekInt,(FuMagRefDataInterp),'x');
% title('Magnitude');
%% Doplneni zadanych dat relevantnimi hodnotami na pozadovanou velikost
% tak, aby bylo zadani kompletne do Fs/2
DummyVekLow = ones(1,FrqStepsCntLow - 1); % pomocne vektory
DummyVekHigh = ones(1,FrqStepsCntUpp - 1);
% Nasleduje sestaveni vektoru
% pred zadane hodnoty se vlozi rada prvni hodnoty zadanych dat
% za zadane hodnoty se vlozi rada posledni hodnoty zadanych dat
FuMagRefOrig = [DummyVekLow.*FuMagRefDataInterp(1)...
    FuMagRefDataInterp...
    DummyVekHigh.*FuMagRefDataInterp(length(FuMagRefDataInterp))];
FuPhaseRefOrig = [DummyVekLow.*FuPhaseRefDataInterp(1)...
    FuPhaseRefDataInterp...
    DummyVekHigh.*FuPhaseRefDataInterp(length(FuPhaseRefDataInterp))];
FuMagRef = FuMagRefOrig;
FuPhaseRef = FuPhaseRefOrig;
%% Doplneni zadani o kompenzaci vlivu doplnkovych filtru a zpozdeni v DSP
Rf1 = 289; % parametry filtru
Rf2 = 47000;

```



```

Cf = 2.2e-8;
FuAntial = Rf2/(Rf1*Rf2*Cf*s + Rf1 + Rf2); % prenos doplnkovych filtru
[magAntial,phaseAntial,wout] = bode(FuAntial,Wvek); % vypočet frek. průběhu
magAntial1 = reshape(magAntial,length(wout),1); % dopln. filtru
phaseAntial1 = reshape(phaseAntial,length(wout),1);
FuDelay = tf([1],[1 0],T); % model zpoždění o jeden vzorek
[magDelay,phaseDelay,wout] = bode(FuDelay,Wvek); % vypočet frek. průběhu
magDelay1 = reshape(magDelay,length(wout),1); % zpoždění vypočtu
phaseDelay1 = reshape(phaseDelay,length(wout),1);
magAdd = (magAntial1.*magDelay1)'; % spojení vlivu zpoždění a filtru ampl.
phaseAddDeg = (phaseDelay1+ phaseAntial1)'; % a fáze
phaseAdd = pi.*phaseAddDeg./180; % přepočet na radiány
% fáze kompenzována jen v oblasti zájmu
phaseAdd(1,MaxIndexInt:K) = phaseAdd(MaxIndexInt-1);
%% Pro odstranění kompenzace amplitudy zakomentovat následující řádek
% FuMagRef = FuMagRef./magAdd;
%% Pro odstranění kompenzace fáze zakomentovat následující řádek
% FuPhaseRef = FuPhaseRef - phaseAdd;
%%
for a = 1:1:length(FuPhaseRef) % Uprava fáze, přesune ji do rozsahu +-pi
    periods = floor(abs(FuPhaseRef(a))/pi); % Počet celých násobků pi
    FuPhaseRef(a) = FuPhaseRef(a) - sign(FuPhaseRef(a))*...
        (periods+mod(periods,2))*pi; % Hodnota v pozadí rozsahu
end
%% Hledání filtru pomocí numerické optimalizace
% Optimalizace, hledání jen podle x - vektor polynomu
[x fval] = fminsearch(@(x) FIRcritFvek(x,...
    M,Fdvek,K,FuMagRef,FuPhaseRef,...
    PsiMtx,FiMtx),x0,...
    optimset('MaxFunEvals',30000))
FuFilterDen = zeros(1,M + 1); % vektory cit. a jm. pro přípravu
FuFilterNum = x; % dat k odeslání
FuFilterDen(1) = 1; % vektor pro jmenovatele
FuFilter = tf(x,FuFilterDen,T) % vytvoření přenosu daného optimalizací
figure; % pomocné zobrazení nul navrženého filtru
pzplot(FuFilter);
%% Vypočet charakteristik navrhovaného filtru
[magFilter,phaseFilter,wout] = bode(FuFilter,Wvek); % vypočet frek. průběhu
magFilter1 = reshape(magFilter,length(wout),1).*magAdd'; % převedení vektoru
phaseFilterNoPar1 = reshape(phaseFilter,length(wout),1); % do spr. tvaru
phaseFilter1 = phaseFilterNoPar1 + phaseAddDeg'; % s doplněním o vliv
% přídavných filtru a zpoždění vypočtu
phaseFilter2 = zeros(length(phaseFilter1),1);
for a = 1:1:length(phaseFilter1) % Uprava fáze do rozsahu +-180
    periods = floor(abs(phaseFilter1(a))/180); % Počet celých násobků 180
    phaseFilter2(a) = phaseFilter1(a) ...
        - sign(phaseFilter1(a))*(periods+mod(periods,2))*180;
end
phaseFilterNoPar2 = zeros(length(phaseFilterNoPar1),1);
for a = 1:1:length(phaseFilterNoPar1) % Uprava fáze do rozsahu +-180
    periods = floor(abs(phaseFilterNoPar1(a))/180); % Počet celých násobků
    phaseFilterNoPar2(a) = phaseFilterNoPar1(a) ...
        - sign(phaseFilterNoPar1(a))*(periods+mod(periods,2))*180;
end
%% Vytvoření promenných pro odeslání do ARMu
% Vytvoření dvou řádkových vektorů, dlouhých 10.
% Jeden pro citatele, jeden pro jmenovatele.
% Pro požadovanou délku doplněno nulami.
FuFilterDen2Send = single([FuFilterDen zeros(1,10-length(FuFilterDen))]);
FuFilterNum2Send = single([FuFilterNum zeros(1,10-length(FuFilterNum))]);
% Přepočet hodnoty odporu W na index rele výstupu.
% 10 Ohm odpovídá 1
% 100 Ohm 2
% 1000 Ohm 3
% 10000 Ohm 4

```



```

indexWout = uint8(4);
switch W
    case 10
        indexWout = uint8(1);
    case 100
        indexWout = uint8(2);
    case 1000
        indexWout = uint8(3);
    case 10000
        indexWout = uint8(4);
    otherwise
        indexWout = uint8(4);
end
%% Zobrazení výsledku
%% Zobrazení průběhu navrhovaného filtru
% Průběhy faze
figure;
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezené pasmo
% 180.*FuPhaseRefOrig(MinIndexInt:MaxIndexInt)./pi,...
% 'red','LineWidth',2); % vykreslí fazi bez kompenzaci, osa x log.
semilogx(Fvek,180.*FuPhaseRefOrig./pi,... % neomezené pasmo
% 'red','LineWidth',2); % vykreslí fazi bez kompenzaci, osa x log.
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezené pasmo
% 180.*FuPhaseRef(MinIndexInt:MaxIndexInt)./pi,
% 'cyan','LineWidth',2);
semilogx(Fvek,180.*FuPhaseRef./pi,'cyan','LineWidth',2); % faze s komp.
% neomezené pasmo

% hold on;
% semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezené pasmo
% phaseFilterNoPar2(MinIndexInt:MaxIndexInt),...% faze bez dodat
% 'blue','LineWidth',2); % filtru a zpozd.
hold on;
semilogx(Fvek,phaseFilterNoPar2,'blue','LineWidth',2); % syntetizovaná faze
% hold on; % neomezené pasmo
% semilogx(Fvek(MinIndexInt:MaxIndexInt),...
% phaseFilter2(MinIndexInt:MaxIndexInt),...
% 'green','LineWidth',2);
semilogx(Fvek,phaseFilter2,'green','LineWidth',2);
set(gca,'FontSize',15); % nastavení velikosti čísel grafu
xlabel('Frekvence [Hz]','fontSize',15); % popiska osy x
ylabel('Fáze [°]','fontSize',15); % popiska osy y
title({'Fáze filtru požadovaná versus syntetizovaná'},...
'fontSize',15); % nadpis grafu
grid on; % zapne mřížku
legend('Zadaná','Zadaná s korekcí',...
'Synt. bez parazit vliv.','Syntetizovaná',...
'Location','Best'); % legenda grafu

% Průběhy absolutní hodnoty
figure % nové okno pro graf
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezené pasmo, zadání abs. h.
(FuMagRefOrig(MinIndexInt:MaxIndexInt)), 'red','LineWidth',2);
semilogx(Fvek,(FuMagRefOrig),'red','LineWidth',2); % neomezené pasmo
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezené pasmo syntet. amp.
% (magFilter1(MinIndexInt:MaxIndexInt)),...
% 'blue','LineWidth',2);
semilogx(Fvek,(magFilter1),'blue','LineWidth',2); % neomezené pasmo
set(gca,'FontSize',15); % nastavení velikosti čísel grafu
xlabel('Frekvence [Hz]','fontSize',15); % popiska osy x
ylabel('Absolutní hodnota','fontSize',15); % popiska osy y
title({'Absolutní hodnota filtru požadovaná versus syntetizovaná'},...
'fontSize',15); % nadpis grafu
legend('Zadaná','Syntetizovaná','Location','Best'); % legenda grafu
grid on; % zapne mřížku

```



```

%% Zobrazení výsledných průběhu implementované Z
ZmagDes = W./magFilter1; % Prepocet na průběhy impedance
ZphaseDes = (-1)*phaseFilter2;
% Zobrazení fází
figure;
semilogx(FvekData,ZPhaseRefData,'red','LineWidth',2); % fáze zadání
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),...
    ZphaseDes(MinIndexInt:MaxIndexInt),...
    'blue','LineWidth',2); % syntetizovaná fáze
set(gca,'FontSize',15); % nastavení velikosti čísel grafu
set(gca,'XLim',[FrqIntMin FrqIntMax]); % limity osy x
xlabel('Frekvence [Hz]','fontSize',15); % popiska osy x
ylabel('Fáze [°]','fontSize',15); % popiska osy y
title({'Fáze impedance požadovaná versus emulovaná'},...
    'fontSize',15); % nadpis grafu
grid on; % zapne mřížku
legend('Zadaná','Syntetizovaná','Location','Best'); % legenda grafu
% Zobrazení amplitud
figure % nové okno pro graf
semilogx(FvekData,(ZMagRefData),'red','LineWidth',2); % amplituda zadání
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),...
    ZmagDes(MinIndexInt:MaxIndexInt),...
    'blue','LineWidth',2); % syntetizovaná amplituda
set(gca,'FontSize',15); % nastavení velikosti čísel grafu
set(gca,'XLim',[FrqIntMin FrqIntMax]); % meze osy x
xlabel('Frekvence [Hz]','fontSize',15); % popiska osy x
ylabel('Absolutní hodnota [Ω]','fontSize',15); % popiska osy y
title({'Absolutní hodnota impedance požadovaná versus emulovaná'},...
    'fontSize',15); % nadpis grafu
legend('Zadaná','Syntetizovaná','Location','Best'); % legenda grafu
grid on; % zapne mřížku

```

### Výpis skriptu NavrhFiltruIIR.m

```

close all;
clear all;
clc;
%% Nastavení základních veličin
W = 100; % W odpor určující rozsah absolutní hodnoty
Fs = 100000; % Vzorkovací frekvence a perioda
T = 1/Fs;
z = tf('z',T); % z jako proměnná pro diskretní přenosové funkce
s = tf('s'); % s jako proměnná pro spojitý přenosové funkce
MarkDataRead = 0; % Marker, že data nebyla načtena ze souboru
%% Zadání radu filtru
N = 3; % N jmenovatel
M = N; % Rad filtru M čitatel, možno změnit dle potřeby
%% Načtení dat z tabulkového souboru
roughData = xlsread('data.xlsx'); % čtení z excel souboru
% Vyčtení dat - závisle na struktuře souboru, v případě
% potřeby nutno upravit
FvekData = roughData(:,1); % Vyčtení sloupce, 1. sloupec frekvence
Phase0 = roughData(:,2); % 2. sloupec fáze 0
Phase1 = roughData(:,3); % 3. sloupec fáze 1
Mag1 = roughData(:,4); % 4. sloupec Magnituda 1
Mag0 = roughData(:,5); % 5. sloupec Magnituda 0
ZMagRefData = 20.*Mag0; % Data brána jako zadání mag a fáze Z
ZPhaseRefData = Phase0;
FuMagRefData = (W./ZMagRefData); % Prepocet průběhu fáze a magnitudy Z
FuPhaseRefDegData = ((-1).*ZPhaseRefData); % na průběh filtru Fu
FuPhaseRefRadData = (pi/180).*FuPhaseRefDegData; % Prepocet fáze na radiány
MarkDataRead = 1; % Marker, že data byla načtena ze souboru
%% Uprava zadání dodána později jako pokus

```



```

% indexMidd = find(FvekData > 1100,1); % Nalezení prostredku okolo ktereho
% FrqMidd = FvekData(indexMidd); % se bude otacet
% for a = 1 : indexMidd
%     FuMagRefDataNew(a) = FuMagRefData(a);
%     FvekDataNew(a) = FvekData(a);
% end
% for a = indexMidd + 1: length(FuMagRefData)
%     FuMagRefDataNew(a) = FuMagRefData(indexMidd)+...
%         FuMagRefData(indexMidd)...
%         -FuMagRefData(indexMidd-(a-indexMidd));
%     FvekDataNew(a) = FvekData(a);
% end
% FvekData = FvekDataNew;
% FuMagRefData = FuMagRefDataNew;
% figure; % Pomocne zobrazeni upravenych dat, jen pro ladeni
% semilogx(FvekDataNew,FuMagRefDataNew);
% hold on;
% semilogx(FvekData,FuMagRefData,'red');
%% Parametry frekvencni charakteristiky
FrqMin = 10; % Nej nizsi uvazovana frekvence
FrqIntMin = 10 % Dolni mez oblasti zajmu ve frekvenci, default. hodnota
FrqIntMax = 5000; % Horni mez oblasti zajmu ve frekvenci, default. hodnota
FrqMax = Fs/2; % Nejvyssi uvazovana frekvence je Nyquistova
if(MarkDataRead == 1) % Pokud byla data uspesne nactena, jinak default
    FrqIntMin = FvekData(1);
    FrqIntMax = FvekData(length(FvekData));
end
FrqStepsCntLow = 50; % Pocet kroku ve frekvenci pod oblasti zajmu
FrqStepsCntInt = 500; % Pocet kroku ve frekvenci v oblasti zajmu
FrqStepsCntUpp = 100; % Pocet kroku ve frekvenci nad oblasti zajmu
MinIndexInt = FrqStepsCntLow; % Pomocne indexy pro zobrazovani
MaxIndexInt = FrqStepsCntLow + FrqStepsCntInt - 1;
% Dilci vektory frekvence pro jednotlivé casti
FvekLow = linspace(FrqMin,FrqIntMin,FrqStepsCntLow);
FvekInt = linspace(FrqIntMin,FrqIntMax,FrqStepsCntInt);
FvekUpp = linspace(FrqIntMax,FrqMax,FrqStepsCntUpp);
% Celkovy vektor frekvenci [Hz]
Fvek = [FvekLow(1:FrqStepsCntLow-1) FvekInt FvekUpp(2:FrqStepsCntUpp)];
Wvek = 2*pi.*Fvek; % Celkovy vektor kruh frekvenci [rad.s^-1]
Fdek = Wvek./Fs; % Vektor digitalnich frekvenci [1]
K = length(Fvek); % Pocet bodu frek. charakteristiky
%% Nastaveni parametru optimalizace - pocatecni bod a vahove matice
% Nacte ulozone koeficienty filtru
x0rough = load('x0pokusProText2013_12_29.mat');
x0 = x0rough.x; % pouzije je jako vychodi bod optimalizace
tempVekNum = RandInitPolyDesign(M); % Navrhne koeficienty citatele nahodne
tempVekDen = RandInitPolyDesign(N); % Vytvori koeficienty jmenov. nahodne
if(length(x0) ~= M+N+1) % pokud nactenou hodnotu nelze pouzit
    disp('Nesouhlasí délka x0 a x, nutno vygenerovat x0!'); % upozorni
    x0 = [tempVekNum tempVekDen(2:length(tempVekDen))]; % a pouzije nahodne
    % vygenerovane h.
end
%% Nasledujici radek zakomentovat pokud se pro x0 pouziva nactena hodnota
% ze souboru; pokud není je pro x0 pouzita nahodna hodnota
% x0 = [tempVekNum tempVekDen(2:length(tempVekDen))]; % Slozi cit. a jmen.
%%
PsiMtx = eye(K); % Vahova matice pro optimalizaci, PsiMtx pro magnitudu
FiMtx = eye(K); % Vahova matice pro optimalizaci, FiMtx pro fazi
%% Linearni koeficienty
alfa = 10000; % Hodnoty pro tvorbu vah, alfa pro magnitudu
beta = 1000; % beta pro fazi
% Nasledujici forcyklus vytvori vahy ve tvaru klesajici lin
% zavislosti v oblasti zajmu, jinde je hodnota rovna 1
for a = MinIndexInt : MaxIndexInt % Zdurazneni hodnot ve vah. maticich
    PsiMtx(a,a) = ((alfa - 1)/...
        (MinIndexInt - MaxIndexInt))*(a - MinIndexInt) + 2*alfa;
end

```



```

FiMtx(a,a) = ((beta - 1)/...
              (MinIndexInt - MaxIndexInt))*(a - MinIndexInt) + 2*beta;
end
%% Okno
% MinIndexWindow = find(Fvek > 200,1); % pocatek okna, index dle frekvence
% MaxIndexWindow = find(Fvek > 5000,1); % konec okna
% WindowLength = MaxIndexWindow - MinIndexWindow; % delka okna
% WindowVek = bohmanwin(WindowLength); % volba okna, mozno dat jine dle
% % WindowVek = rectwin(WindowLength); % napovedy
% alfa = 10000; % Hodnoty pro tvorbu vah, alfa pro magnitudu
% beta = 10000; % beta pro fazi
% % Nasledujici forcyklus vytvori vahy ve dle okenkovaciho
% % vektoru v rozsahu okna zadaniem vyse
% for a = MinIndexWindow : MaxIndexWindow - 1
%     PsiMtx(a,a) = PsiMtx(a,a) + WindowVek(a-MinIndexWindow+1)*alfa;
%     FiMtx(a,a) = FiMtx(a,a) + WindowVek(a-MinIndexWindow+1)*beta;
% end
% % UkoefVek = (diag(PsiMtx))'; % Pomocne vektory diagonalnich prvku matic
% % WkoefVek = (diag(FiMtx))'; % PsiMtx a FiMtx, slouzi jen pro zobrazeni
% % figure; % Pouze pomocne zobrazeni prubehu vahovych koeficientu
% % semilogx(Fvek,UkoefVek,'red',Fvek,WkoefVek,'blue');
%% Interpolace dat splajnem pro moznost ruzneho kroku dat a vypoctu
FuPhaseRefDataInterp = spline(FvekData,FuPhaseRefRadData,FvekInt); % faze
FuMagRefDataInterp = spline(FvekData,FuMagRefData,FvekInt); % amplituda
%Pomocne zobrazeni dat po interpolaci
% figure;
% semilogx(FvekData,FuPhaseRefRadData,...
%           'red',FvekInt,FuPhaseRefDataInterp,'x');
% title('Phase');
% figure;
% semilogx(FvekData,(FuMagRefData),
%           'red',FvekInt,(FuMagRefDataInterp),'x');
% title('Magnitude');
%% Doplneni zadanych dat relevantnimi hodnotami na pozadovanou velikost
% tak, aby bylo zadani kompletne do Fs/2
DummyVekLow = ones(1,FrqStepsCntLow - 1); % pomocne vektory
DummyVekHigh = ones(1,FrqStepsCntUpp - 1);
% Nasleduje sestaveni vektoru
% pred zadane hodnoty se vlozi rada prvni hodnoty zadanych dat
% za zadane hodnoty se vlozi rada posledni hodnoty zadanych dat
FuMagRefOrig = [DummyVekLow.*FuMagRefDataInterp(1)...
                FuMagRefDataInterp...
                DummyVekHigh.*FuMagRefDataInterp(length(FuMagRefDataInterp))];
FuPhaseRefOrig = [DummyVekLow.*FuPhaseRefDataInterp(1)...
                  FuPhaseRefDataInterp...
                  DummyVekHigh.*FuPhaseRefDataInterp(length(FuPhaseRefDataInterp))];
FuMagRef = FuMagRefOrig;
FuPhaseRef = FuPhaseRefOrig;
%% Doplneni zadani o kompenzaci vlivu doplnkovych filtru a zpozdeni v DSP
Rf1 = 289; % parametry filtru
Rf2 = 47000;
Cf = 2.2e-8;
FuAntial = Rf2/(Rf1*Rf2*Cf*s + Rf1 + Rf2); % prenos doplnkovych filtru
[magAntial,phaseAntial,wout] = bode(FuAntial,Wvek); % vypočet frek. prubehu
magAntial1 = reshape(magAntial,length(wout),1); % dopln. filtru
phaseAntial1 = reshape(phaseAntial,length(wout),1);
FuDelay = tf([1],[1 0],T); % model zpozdeni o jeden vzorek
[magDelay,phaseDelay,wout] = bode(FuDelay,Wvek); % vypočet frek. prubehu
magDelay1 = reshape(magDelay,length(wout),1); % zpozdeni vypoctu
phaseDelay1 = reshape(phaseDelay,length(wout),1);
magAdd = (magAntial1.*magDelay1)'; % spojeni vlivu zpozdeni a filtru ampl.
phaseAddDeg = (phaseDelay1+ phaseAntial1)'; % a faze
phaseAdd = pi.*phaseAddDeg./180; % prepocet na radiany
% faze kompenzovana jen v oblasti zajmu
phaseAdd(1,MaxIndexInt:K) = phaseAdd(MaxIndexInt-1);

```



```

%% Pro odstranění kompenzace amplitudy zakomentovat tento radek
% FuMagRef = FuMagRef./magAdd;
%% Pro odstranění kompenzace fáze zakomentovat tento radek
FuPhaseRef = FuPhaseRef - phaseAdd;
%%
for a = 1:length(FuPhaseRef) % Uprava fáze do rozsahu -pi +pi
    periods = floor(abs(FuPhaseRef(a))/pi);
    FuPhaseRef(a) = FuPhaseRef(a) ...
        - sign(FuPhaseRef(a)) * (periods+mod(periods,2)) * pi;
end
%% Hledání filtru pomocí numerické optimalizace
% Optimalizace, hledání jen podle x
lb = -inf.*ones(1,M + 1 + N); % dolní mez x - praktické -nekonečno
ub = +inf.*ones(1,M + 1 + N); % horní mez y - praktické nekonečno
options=optimset('Algorithm','sqp', ... % parametry - metoda
    'MaxFunEvals',30000, ... % limit evaluací
    'MaxIter',1000, ... % limit iterací
    'TolFun',1e-9, ... % nejmenší krok funkční hodnoty
    'TolX',1e-6, ... % nejmenší krok x
    'TolCon',1e-10); % nejmenší krok omezující fce
[x,fval,exitflag] = fmincon(@x ... % optimalizace dle proměnné x
    IIRcritFvek(x,M,N,Fdvek,K,FuMagRef,FuPhaseRef,PsiMtx,FiMtx) ...
    ,x0,[],[],[],[],lb,ub, ... % lineární omezení není
    @x IIRrootsConstrain(x,M,N),options) % nelineární omezení fci
if (exitflag < 1) % pokud je výsledek nestabilní zastaví skript
    error('Optimalizace neproběhla správně! Výsledek je chybný!');
end
FuFilterDen = [1 x(M+2:M+N+1)]; % vytvoření vektoru čitatele a jmen.
FuFilterNum = [x(1:M+1) zeros(1,N-M)]; % pokud N != M doplněno nulami zpredu
    % protože MATLAB počítá s kladnými
    % mocninami z
FuFilter = tf(FuFilterNum,FuFilterDen,T) % přenos dany optimalizací
figure;
pzplot(FuFilter); % zobrazení nul a polů navrženého přenosu
%% Vytvoření proměnných pro odeslání do ARMu
% Vytvoření dvou řádkových vektorů, dlouhých 10.
% Jeden pro čitatele, jeden pro jmenovatele.
% Pro požadovanou délku doplněno nulami.
FuFilterDen2Send = single([FuFilterDen zeros(1,10-length(FuFilterDen))]);
FuFilterNum2Send = single([FuFilterNum zeros(1,N-M) ...
    zeros(1,10-length(FuFilterNum)-(N-M))]);
    % Přepočít hodnoty odporu W na index rele výstupu.
    % 10 Ohm odpovídá 1
    % 100 Ohm 2
    % 1000 Ohm 3
    % 10000 Ohm 4
indexWout = uint8(4);
switch W
    case 10
        indexWout = uint8(1);
    case 100
        indexWout = uint8(2);
    case 1000
        indexWout = uint8(3);
    case 10000
        indexWout = uint8(4);
    otherwise
        indexWout = uint8(4);
end
%% Výpočet charakteristik navrženého filtru
[magFilter,phaseFilter,wout] = bode(FuFilter,Wvek); % výpočet frek. průběhu
magFilter1 = reshape(magFilter,length(wout),1).*magAdd'; % převod vektorů
phaseFilterNoPar1 = reshape(phaseFilter,length(wout),1); % do spr. tvaru
phaseFilter1 = phaseFilterNoPar1 + phaseAddDeg'; % s doplněním o vliv
    % přídavných filtrů a zpoždění výpočtu

```





```

phaseFilter2 = zeros(length(phaseFilter1),1);
for a = 1:1:length(phaseFilter1) % Uprava faze do rozsahu +-180
    periods = floor(abs(phaseFilter1(a))/180); % Pocet celych nasobku 180
    phaseFilter2(a) = phaseFilter1(a) ...
        - sign(phaseFilter1(a))*(periods+mod(periods,2))*180;
end
phaseFilterNoPar2 = zeros(length(phaseFilterNoPar1),1);
for a = 1:1:length(phaseFilterNoPar1) % Uprava faze do rozsahu +-180
    periods = floor(abs(phaseFilterNoPar1(a))/180); % Pocet celych nasobku
    phaseFilterNoPar2(a) = phaseFilterNoPar1(a)...
        - sign(phaseFilterNoPar1(a))*(periods+mod(periods,2))*180;
end
%% Zobrazeni prubehu navrhovaneho filtru
% Prubehy faze
figure;
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezene pasmo
%     180.*FuPhaseRefOrig(MinIndexInt:MaxIndexInt)./pi,...
%     'red','LineWidth',2); % vykresli fazi bez kompenzaci, osa x log.
semilogx(Fvek,180.*FuPhaseRefOrig./pi,... % neomezene pasmo
%     'red','LineWidth',2); % vykresli fazi bez kompenzaci, osa x log.
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezene pasmo
%     180.*FuPhaseRef(MinIndexInt:MaxIndexInt)./pi,
%     'cyan','LineWidth',2);
semilogx(Fvek,180.*FuPhaseRef./pi,'cyan','LineWidth',2); % faze s komp.
% neomezene pasmo
% hold on;
% semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezene pasmo
%     phaseFilterNoPar2(MinIndexInt:MaxIndexInt),...% faze bez dodat
%     'blue','LineWidth',2); % filtru a zpozd.
hold on;
semilogx(Fvek,phaseFilterNoPar2,'blue','LineWidth',2); % syntetizovana faze
% hold on; % neomezene pasmo
% semilogx(Fvek(MinIndexInt:MaxIndexInt),...
%     phaseFilter2(MinIndexInt:MaxIndexInt),...
%     'green','LineWidth',2);
semilogx(Fvek,phaseFilter2,'green','LineWidth',2);
set(gca,'FontSize',15); % nastaveni velikosti cislic grafu
xlabel('Frekvence [Hz]','fontsize',15); % popiska osy x
ylabel('Fáze [°]','fontsize',15); % popiska osy y
title({'Fáze filtru požadovaná versus syntetizovaná'},...
'fontsize',15); % nadpis grafu
grid on; % zapne mřížku
legend('Zadaná','Zadaná s korekcí',...
'Synt. bez parazit vliv.','Syntetizovaná',...
'Location','Best'); % legenda grafu
% Prubehy absolutni hodnoty
figure % nove okno pro graf
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezene pasmo, zadani abs. h.
(FuMagRefOrig(MinIndexInt:MaxIndexInt)),'red','LineWidth',2);
semilogx(Fvek,(FuMagRefOrig),'red','LineWidth',2); % neomezene pasmo
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),... % omezene pasmo syntet. amp.
%     (magFilter1(MinIndexInt:MaxIndexInt)),...
%     'blue','LineWidth',2);
semilogx(Fvek,(magFilter1),'blue','LineWidth',2); % neomezene pasmo
set(gca,'FontSize',15); % nastaveni velikosti cislic grafu
xlabel('Frekvence [Hz]','fontsize',15); % popiska osy x
ylabel('Absolutní hodnota','fontsize',15); % popiska osy y
title({'Absolutní hodnota filtru požadovaná versus syntetizovaná'},...
'fontsize',15); % nadpis grafu
legend('Zadaná','Syntetizovaná','Location','Best'); % legenda grafu
grid on; % zapne mřížku
%% Zobrazeni vyslednych prubehu implementovane Z
ZmagDes = W./magFilter1; % Prepocet na prubehy impedance

```





```

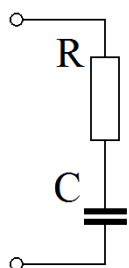
ZphaseDes = (-1)*phaseFilter2;
% Zobrazeni fazi
figure;
semilogx(FvekData,ZPhaseRefData,'red','LineWidth',2); % faze zadani
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),...
    ZphaseDes(MinIndexInt:MaxIndexInt),...
    'blue','LineWidth',2); % syntetiovana faze
set(gca,'FontSize',15); % nastaveni velikosti cislic grafu
set(gca,'XLim',[FrqIntMin FrqIntMax]); % limity osy x
xlabel('Frekvence [Hz]','fontsize',15); % popiska osy x
ylabel('Fáze [°]','fontsize',15); % popiska osy y
title({'Fáze impedance požadovaná versus emulovaná'},...
    'fontsize',15); % nadpis grafu
grid on; % zapne mrizku
legend('Zadaná','Syntetizovaná','Location','Best'); % legenda grafu
% Zobrazeni amplitud
figure % nove okno pro graf
semilogx(FvekData,(ZMagRefData),'red','LineWidth',2); % amplituda zadani
hold on;
semilogx(Fvek(MinIndexInt:MaxIndexInt),...
    ZmagDes(MinIndexInt:MaxIndexInt),...
    'blue','LineWidth',2); % syntetizovana amplituda
set(gca,'FontSize',15); % nastaveni velikosti cislic grafu
set(gca,'XLim',[FrqIntMin FrqIntMax]); % meze osy x
xlabel('Frekvence [Hz]','fontsize',15); % popiska osy x
ylabel('Absolutní hodnota [\Omega]','fontsize',15); % popiska osy y
title({'Absolutní hodnota impedance požadovaná versus emulovaná'},...
    'fontsize',15); % nadpis grafu
legend('Zadaná','Syntetizovaná','Location','Best'); % legenda grafu
grid on; % zapne mrizku

```



## Příloha H Měření základních impedancí

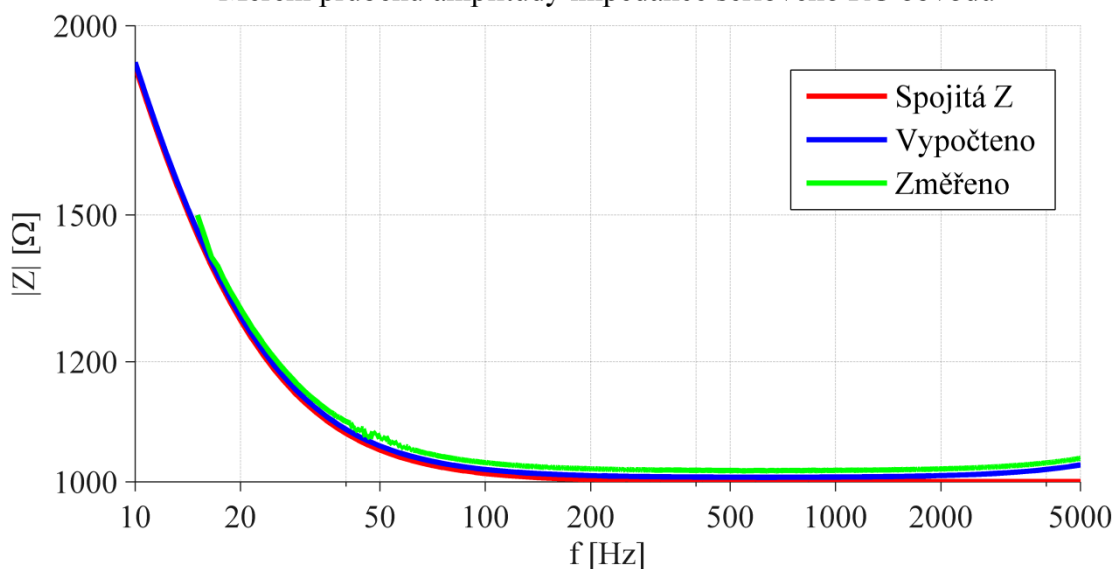
### Sériový RC člen



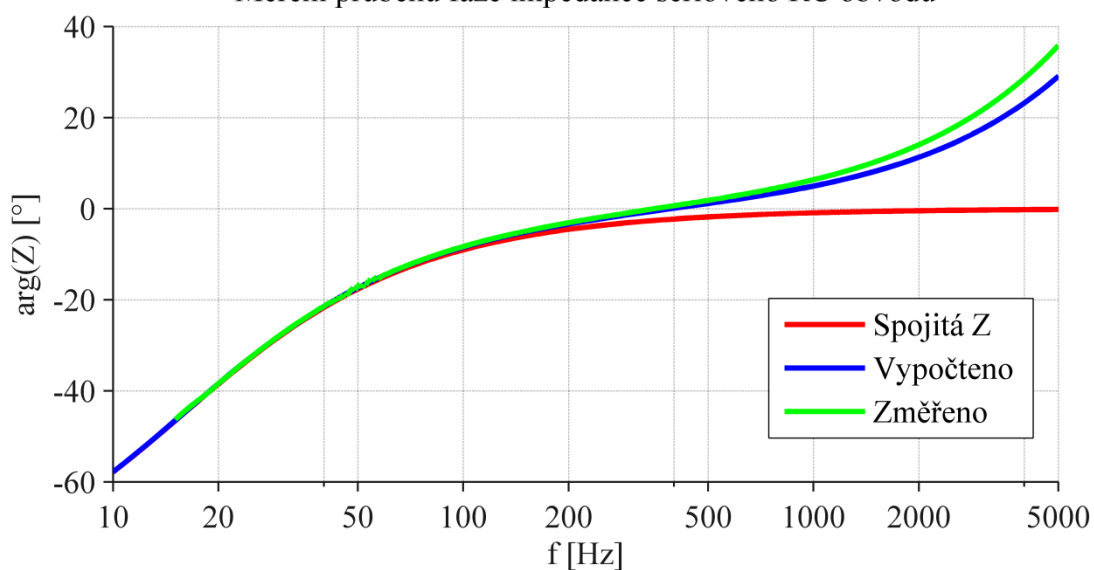
Nastavení:  $W_3 = 1000 \, \Omega$ ,  $f_s = 100 \, \text{kHz}$ ,  $R = 1000 \, \Omega$ , a  $C = 10 \, \mu\text{F}$ ,  
měřeno pomocí analyzátoru HP4195A

$$F(z) = \frac{1 - 0,9995z^{-1}}{1 - 0,999z^{-1}}$$

Měření průběhu amplitudy impedance sériového RC obvodu

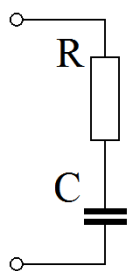


Měření průběhu fáze impedance sériového RC obvodu





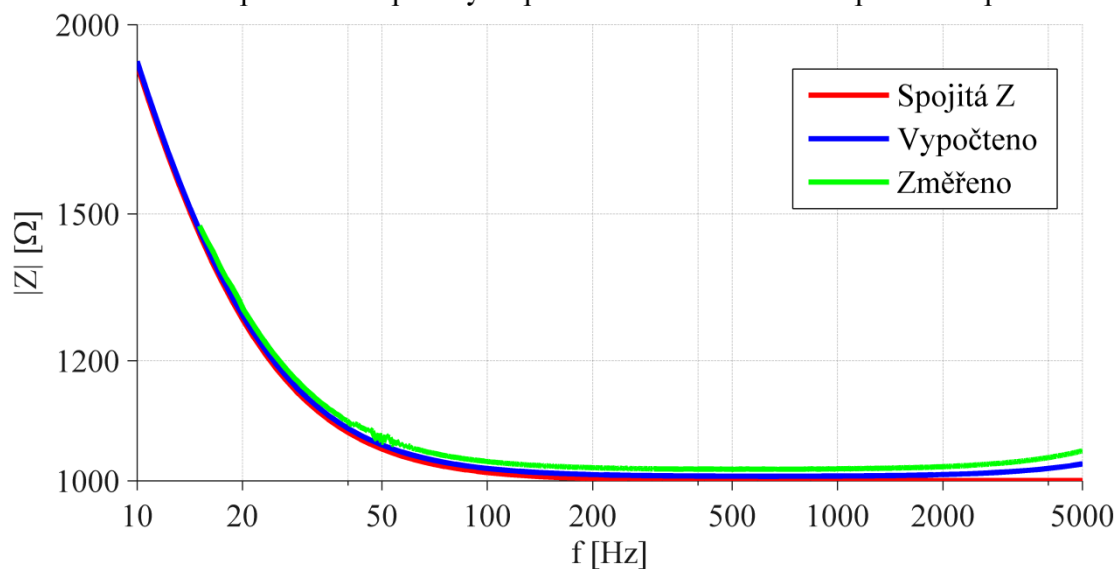
## Sériový RC člen se zápornou kapacitou



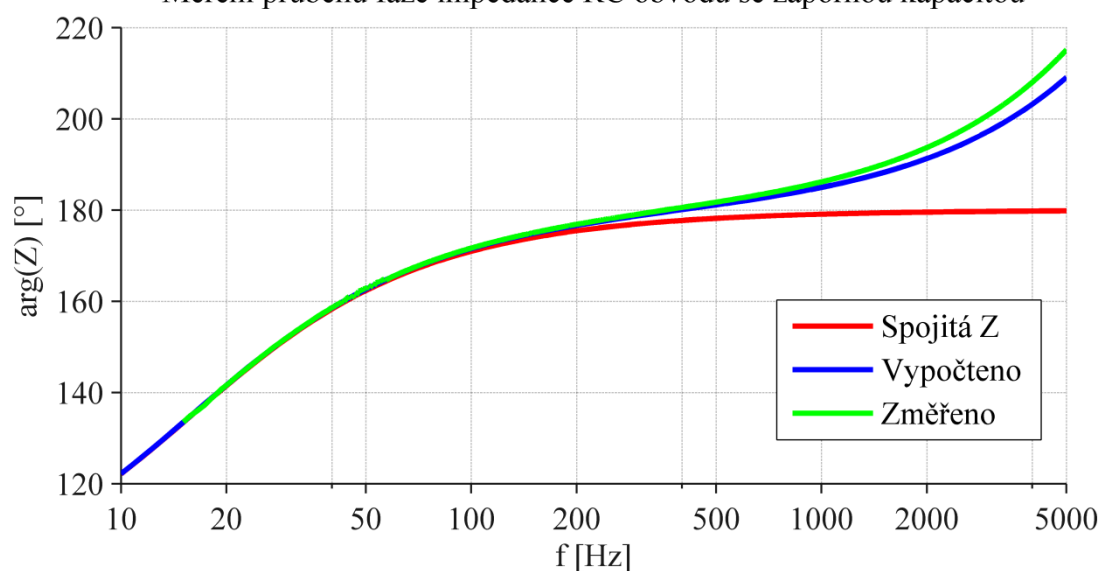
Nastavení:  $W_3 = 1000 \, \Omega$ ,  $f_s = 100 \, \text{kHz}$ ,  $R = -1000 \, \Omega$ ,  
a  $C = -10 \, \mu\text{F}$ , měřeno pomocí analyzátoru HP4195A

$$F(z) = \frac{-1 + 0,9995z^{-1}}{1 - 0,999z^{-1}}$$

Měření průběhu amplitudy impedance RC obvodu se zápornou kapacitou

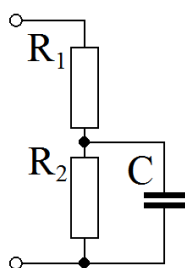


Měření průběhu fáze impedance RC obvodu se zápornou kapacitou





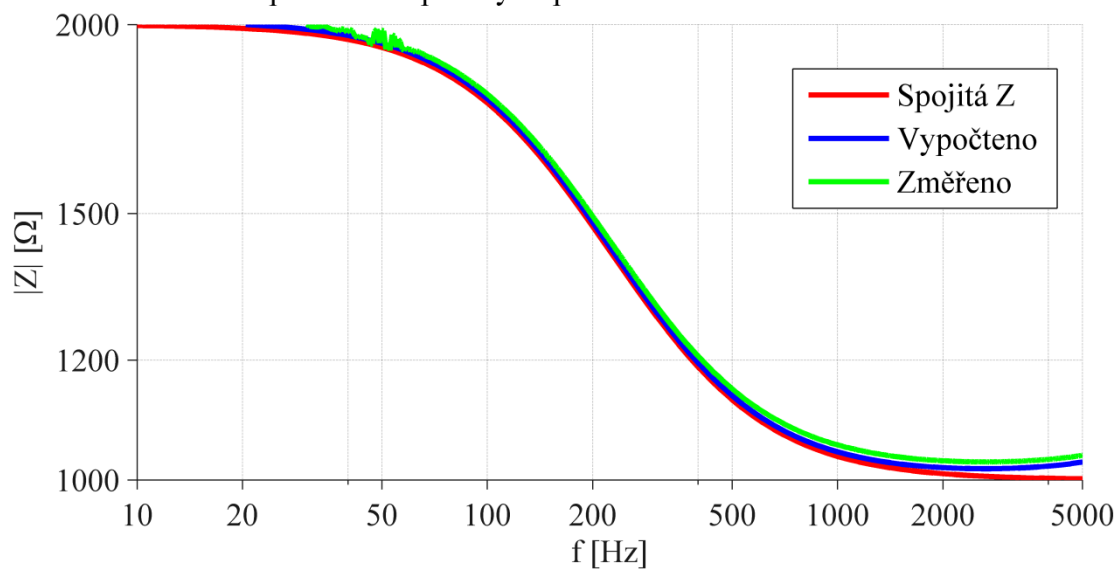
## Kombinovaný RRC člen



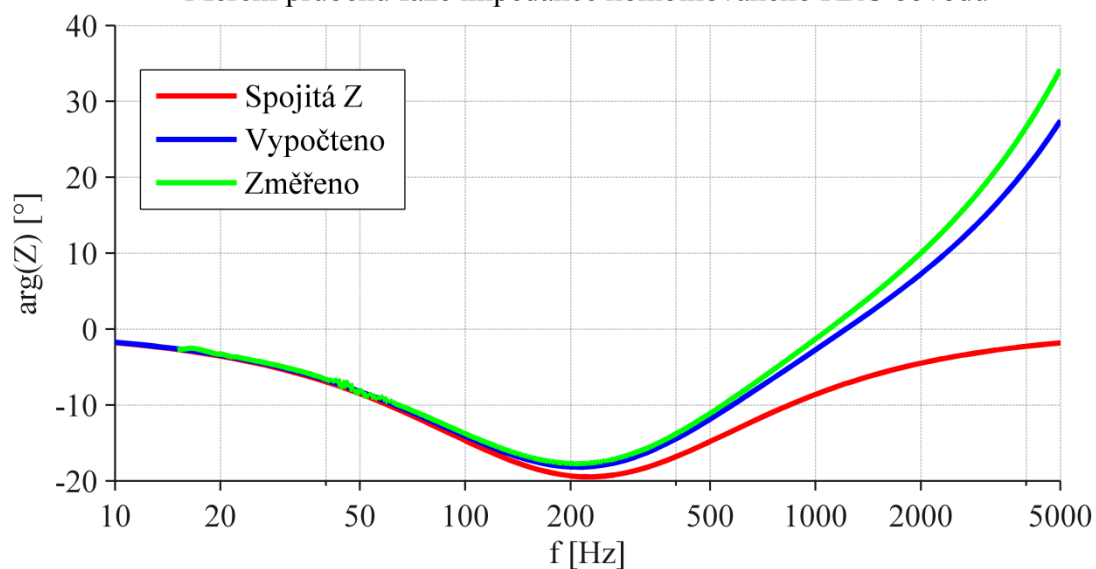
Nastavení:  $W_3 = 1000 \, \Omega$ ,  $f_s = 100 \, \text{kHz}$ ,  $R_1 = 1000 \, \Omega$ ,  $R_2 = 1000 \, \Omega$   
a  $C = 1 \, \mu\text{F}$ , měřeno pomocí analyzátoru HP4195A

$$F(z) = \frac{0,995 - 0,9851z^{-1}}{1 - 0,9802z^{-1}}$$

Měření průběhu amplitudy impedance kombinovaného RRC obvodu

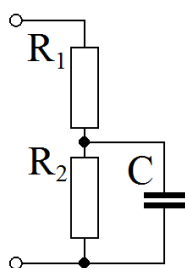


Měření průběhu fáze impedance kombinovaného RRC obvodu





## Kombinovaný RRC člen se zápornou kapacitou

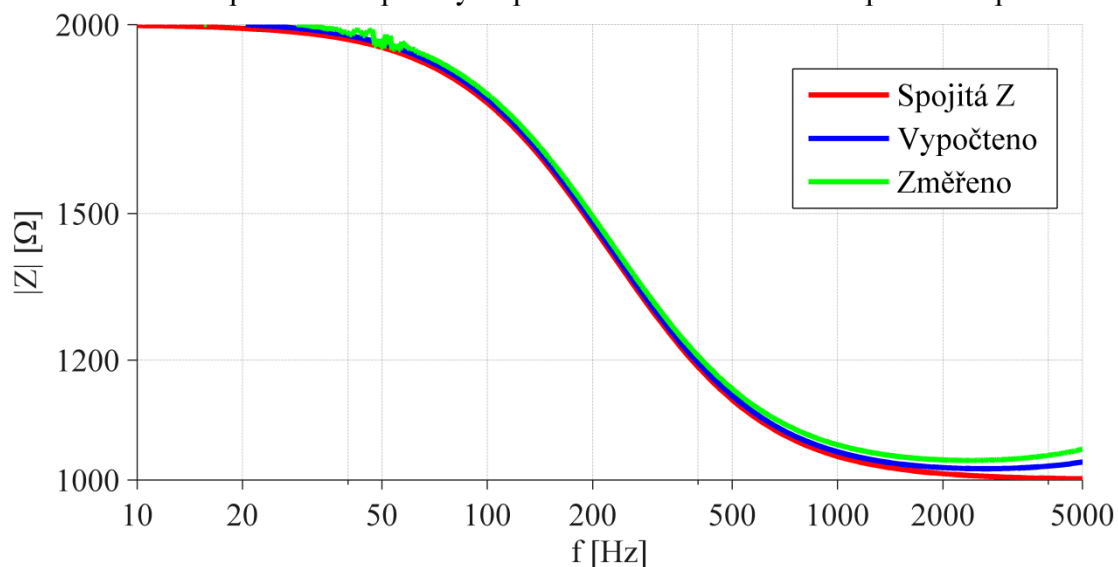


Nastavení:  $W_3 = 1000 \, \Omega$ ,  $f_s = 100 \, \text{kHz}$ ,  $R_1 = -1000 \, \Omega$ ,

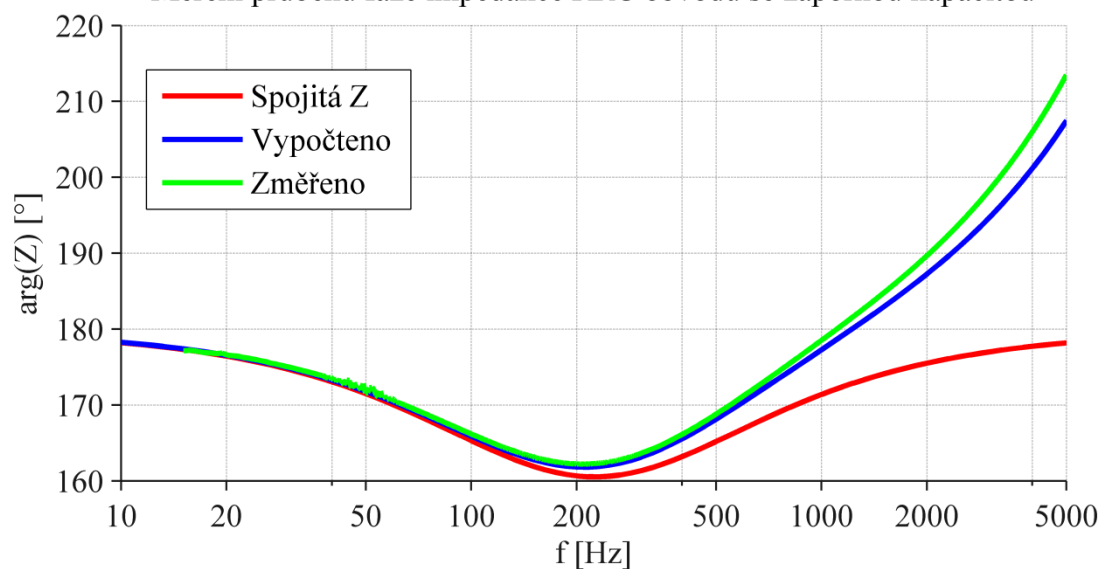
$R_2 = -1000 \, \Omega$  a  $C = -1 \, \mu\text{F}$ , měřeno pomocí analyzátoru HP4195A

$$F(z) = \frac{-0,995 + 0,9851z^{-1}}{1 - 0,9802z^{-1}}$$

Měření průběhu amplitudy impedance RRC obvodu se zápornou kapacitou

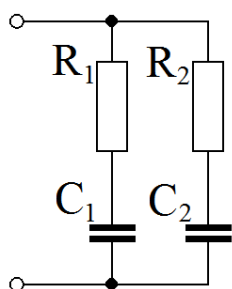


Měření průběhu fáze impedance RRC obvodu se zápornou kapacitou





## Paralelní kombinace dvou sériových RC členů

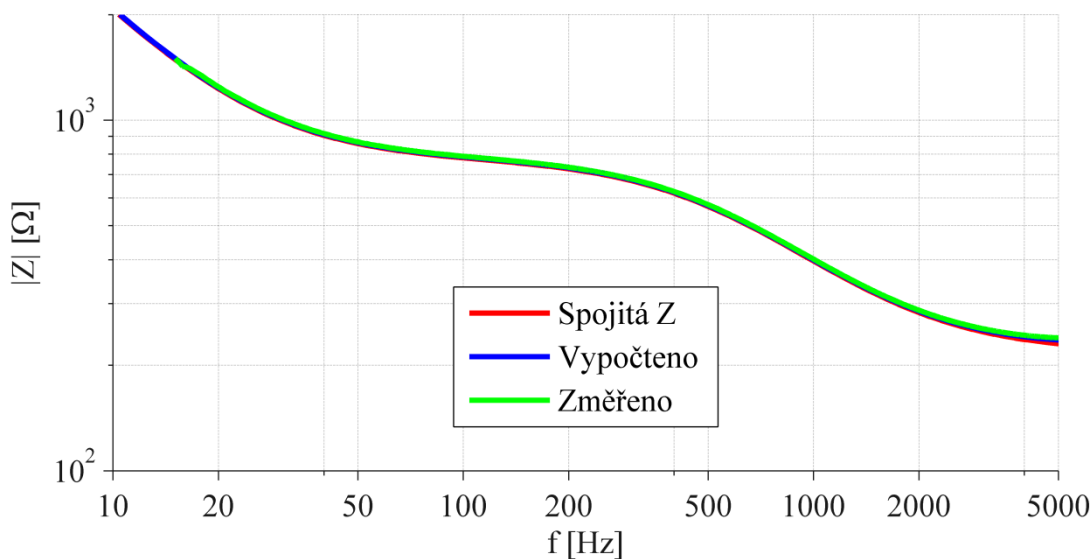


Nastavení:  $W_2 = 100 \, \Omega$ ,  $f_S = 100 \, \text{kHz}$ ,  $R_1 = 800 \, \Omega$ ,  $C_1 = 8 \, \mu\text{F}$

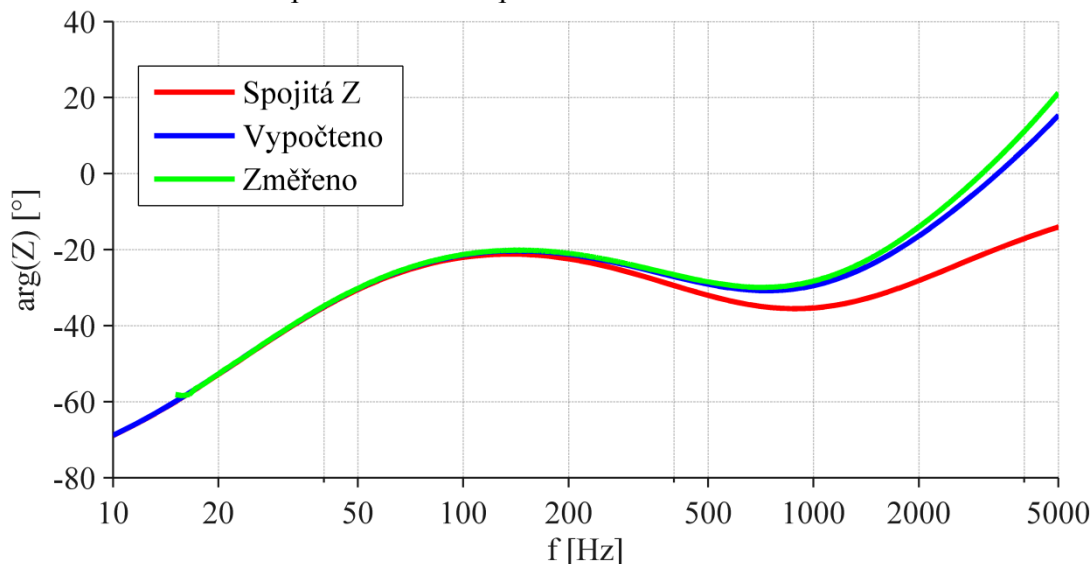
$R_2 = 300 \, \Omega$  a  $C_2 = 300 \, \text{nF}$ , měřeno pomocí analyzátoru HP4195A

$$F(z) = \frac{0,4407 - 0,8677z^{-1} + 0,4271z^{-2}}{1 - 1,893z^{-1} + 0,8933z^{-2}}$$

## Měření průběhu amplitudy impedance kombinace dvou RC obvodů

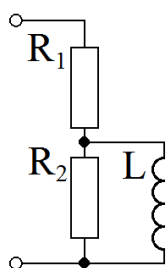


## Měření průběhu fáze impedance kombinace dvou RC obvodů





## Kombinovaný RRL obvod

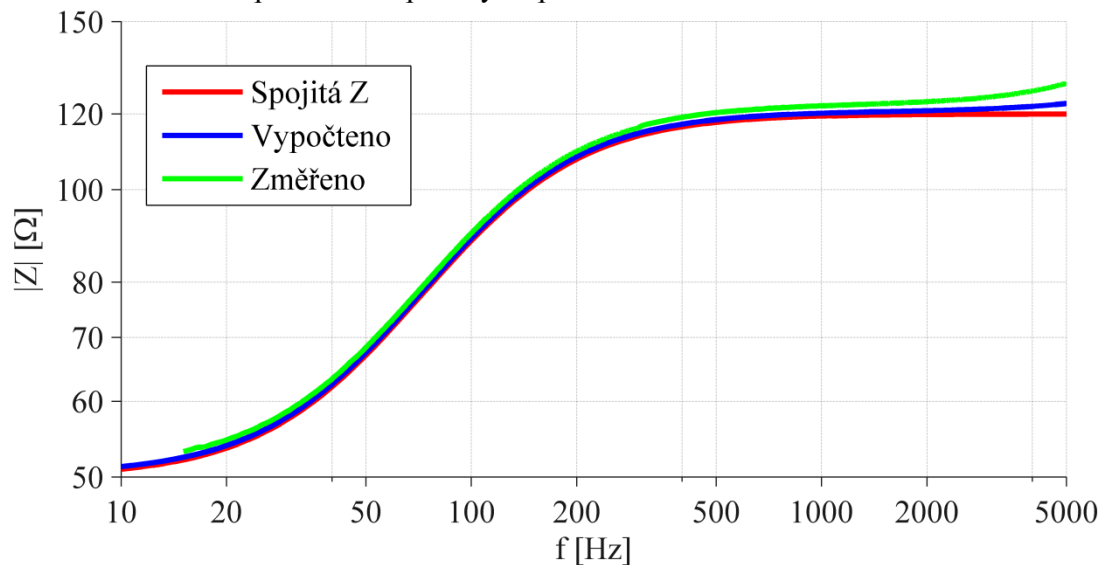


Nastavení:  $W_1 = 10 \, \Omega$ ,  $f_s = 100 \, \text{kHz}$ ,  $R_1 = 50 \, \Omega$ ,

$R_2 = 70 \, \Omega$  a  $L = 0,1 \, \text{H}$ , měřeno pomocí analyzátoru HP4195A

$$F(z) = \frac{0,0835 - 0,08292z^{-1}}{1 - 0,9971z^{-1}}$$

Měření průběhu amplitudy impedance kombinovaného RRL obvodu



Měření průběhu fáze impedance kombinovaného RRL obvodu

